Genome-Wide Association Study Using HapMap Data

Jennifer Eberling

May 5, 2022

Background

Genome-Wide Association Studies (GWAS) are used to help researchers identify parts of DNA that affect a certain trait. They are observational studies that look at the whole genome and then pick out specific locations on the genome that are associated with the specified trait. For this project, the genome is collected by the HapMap Project, and the trait of interest is race.

The HapMap Project is an international tool that was created with the goal of developing a "haplotype map of the human genome" (About the International HapMap Project, 2012). With an abundance of data, the creators wanted to identify patterns in haplotypes that help identify which genes contribute to human health and which genes react to different environmental factors.

There are three genetic terms that are very important to the HapMap project: SNP, genotype, and haplotype. Every person receives two strands of DNA – one from their mother and one from their father. At any given location on the DNA strands, both parents can give a dominant allele (AA), both parents can give a recessive allele (aa), or one parent can give a dominant allele while the other parent gives a recessive allele (Aa). A single nucleotide polymorphism (SNP) is a variant of a specific location on DNA. For example, "rs10868791" is a specific location on a DNA strand. At that location, possible alleles are 'A' and 'G,' so a SNP can look like 'AA,' 'AG,' or 'GG' depending on the person. A genotype is the set of alleles a person has. This may sound similar to the SNP definition, and indeed "genotype" can refer to the "alleles that a person has at a particular SNP" (About the International HapMap Project, 2012). However, a location requires variation in genotypes to qualify as a SNP, so if a location on the DNA strand has the genotype 'AA' for every human on the planet, it is not a SNP. Finally, a haplotype is a collection of SNP alleles in a region of a chromosome that are typically inherited together (Haplotype, 2014). It is a subsection of all genotypes. In summary, a genotype is the

combination of two alleles at a specific location, a SNP is an allele combination that varies, and a haplotype is a specific collection of SNP's.

The HapMap Project contains data on millions of SNP's from hundreds of individuals who reside all over the world. The data used in this paper is a subset that contains 9305 SNP's from 120 individual people along with their race and a unique id number. The 120 people can be split into two distinct groups. 60 of the people are Yaruban (located in Ibadan, Nigeria) and 60 are American (located in Utah, US and with ancestry from Northern and Western Europe) (Haplotype, 2014).

Another concept that will be used in this paper is Hardy-Weinberg Equilibrium. Hardy-Weinberg Equilibrium is the concept that genetic variation will at some point become constant from generation to generation. There are five assumptions made in the calculation of whether a population is in Hardy-Weinberg Equilibrium or not: random mating is present, there is no migration of the population, no genetic mutations occur, no genetic drift occurs, and natural selection is in effect ("Hardy-Weinberg"). Using the available SNP's, this paper will determine whether this population of 120 individuals has reached Hardy-Weinberg Equilibrium.

The goal of this project is to identify SNP's that are associated with human race. To accomplish that, the paper will begin in by filling in missing genotypes with the most popular genotype at that SNP. It will check whether Hardy-Weinberg Equilibrium is met for every SNP, and then make a conclusion about whether it is met for the entire population. Then it will perform logistic regression on race where the p-values are calculated with single variable logistic regression using each SNP as the independent variable one at a time. Lower p-values represent SNP's more highly associated with race. As a final step, the process will be repeated with a data set that has missing values filled in with the SNP's most popular genotype within whichever race the subject with the missing cell has listed. The analyses will be compared to that done with the original methodology to make conclusions about which SNP's are the most associated with race, how missingness affects the analysis, and what proper steps for future research would be.

Methods

In this dataset, there are multiple missing values per person that need to be estimated in order to proceed with the analysis. To fill in the missing SNP locations, the most frequently occurring genotype for each SNP is calculated. To attain the mode information, a loop finds the mode of each column (SNP). Within the first loop is a secondary loop that runs through every row within the column. If a cell has the value NA – meaning that it is missing – then the value is replaced with the mode of the column.

Once the dataset is complete, the second step is to check for Hardy-Weinberg Equilibrium. For each SNP, p and q are calculated where p is the observed probability of the 'A' allele and q is the observed probability of the 'a' allele. To find those values, a loop is created that identifies three scenarios by making a table of all possible alleles and counting how many columns are present. When there are three columns, the genotypes 'AA,' 'Aa,' and 'aa' are all present. Those frequencies are stored as AA, Aa, and aa where the first genotype in the Table is counted as AA, the second as Aa, and the third as aa. The genotypes are then separated into alleles meaning that each two-letter variable becomes two one-letter variables. The pivot_longer() function stacks the alleles while keeping the associated frequencies. Then data is grouped by allele to count how many times 'A' and 'a' are present. p is then calculated by dividing the frequency of 'A' (taken from the summ table) by 240, the number of alleles observed. q is similarly calculated for 'a'. The process is identical for when there are two genotypes.

When only one genotype is present, it is assumed that the genotype is homozygous, meaning it is either 'AA' or 'aa' and not 'Aa' since the probability that all 120 people have 'Aa' is very low given random mating. If all observations are homozygous though, the species may be stuck there with no way to attain a heterozygous genotype until a mutation occurs. With this assumption, all observations are counted and divided by 240 to calculate p, and q is set to 0.

The AA, Aa, and aa frequencies are pulled from the table and put into a vector that is fed into the HWChisq() function from the HardyWeinberg Package using the 'verbose = F' option to suppress extraneous output (Graffelman, 2009). From the function, the p-value is pulled from the test where the null is that the population has reached Hardy-Weinberg Equilibrium and the

alternative is that the population has not. This is run for each SNP and stored in the first row of the empty hwe matrix.

p and q are saved in the matrix as well to periodically check that p^2+2pq+q^2 = 1, confirming our calculation. They are not used for further analysis but could be used in an alternative method to determine if the distribution if the population has reached Hardy-Weinberg Equilibrium.

After the hwe matrix is transposed so that each row is a SNP and the columns are p-value, p, and q, the empty 'group' and 'id' rows are removed. Then an indicator variable is created that is 1 when the Hardy-Weinberg Equilibrium p-value is less than .05 and 0 otherwise. If Hardy-Weingberg Equilibrium is assumed, approximately 5% of the SNP's are expected to be outside of the confidence interval based on random chance alone. If more than 5% of equations are out of the interval, then this population is not in equilibrium. A table of the indicator is piped into prop.table() to get the proportion of significant SNP's.

With the now complete data, logistic regression is used to identify the SNP's most associated with race. varlist and pval_list are created as empty matrices to store the names of all SNP's and the p-values associated with them from simple logistic regression. A loop is set up to run through each SNP listed in varlist. Variables with just one genotype are excluded by an if-statement because you cannot perform simple logistic regression with a single level independent variable.

For variables with at least two levels, the variable is written as a string that is substituted for "%s" in the formulat "group ~ %s" which is stored in code_blueprint. code_blueprint is then substituted into the glm function. These steps are necessary because the glm function does not allow strings to be changed within the function. Storing the variable in code_blueprint every time circumvents that problem. That model is then compared to a basic model with the formula "group ~ 1" via the anova() function. This p-values is stored in the pval_list matrix where the first row is a list of SNP names copied over from varlist and the second row is the p-values for when that SNP alone predicts race.

That matrix of SNP's and p-values is then transposed and made into a data frame with two columns. A third column called manhat is created by taking a negative log of the squared p-value. The manhat column is what will be used in the Manhattan Plots since low p-values will result in proportionally high manhat values. The lowest p-values will become the highest manhat values which will be visible in a Manhattan plot. Using the code

```
P_values %>% filter(is.na(X1)==T)
```

it is found that there are 1659 SNP's that have no p-value, meaning that they had no variation in genotype and thus were not included in the logistic regression loop.

One Manhattan Plot is created first, and then three indicators are specified. The first is 1 when manhat values are above 60. This cutoff is chosen first by looking at the data and finding a cutoff to easily identify only the most significant SNP's. The second indicator is 1 when the p-value is less than 5*10^(-8), a standard cutoff (Zhongsheng, 2021). This will be the cutoff used in the final analysis. The third indicator is slightly more conservative than indicator 2 and is 1 when the p-value is less than 5*10^(-7) and is included for comparison (Zhongsheng, 2021). With these three indicators, three more Manhattan Plots are made with points colored by whether their indicator is 1 or 0.

To search for trends in significant values, a variable called original_order is created with the values 1 through 9305 so that p-values can be plotted against each of the three indicators in another effort to identify trends in significance.

A sensitivity analysis is performed that repeats the above method using a dataset where missing values are calculated differently. Again, an empty 120 x 9307 matrix is created to hold the new values, and a new vector, empty_columns, is created to hold a list of the columns that are empty for every observation in either the CEU or YRI race groups. These columns will be excluded because it is unclear if they are completely different from the other group or the same and the missingness is due to a testing center discrepancy. In a loop, two vectors are created using is.na() to hold a string of which of the 60 values are missing for each race group. 'allmissing' is a vector that holds 60 TRUE values. The setequal() function compares the two

vectors to allmissing, and if that setequal is TRUE, the column number is added to empty_columns to be excluded from the analysis.

HMCEU and HMYRI are matrices that hold the SNP information for each of their respective races. Two separate loops are created to run through every column in each of them. If the column has not been recorded in empty_columns, the mean of that column is calculated. A second loop within the first runs through every row. If the cell is not empty, it stores that value in a new data frame "fullmatgroup". If the cell is empty, it fills in the missing value with the mean for that column within its race. This data is then used to calculate if Hardy-Weinberg Equilibrium is met and what percentage of SNP's are significantly associated with race.

Results

Table 1 shows the proportion of SNP's that show significant evidence of not being in Hardy-Weinberg Equilibrium. The first row is for when the missing values are calculated with the average genotype for the SNP and the second row is for when they are calculated with the group (race) average. Only 8841 SNP's are usable in the second row since SNP's with information missing for an entire race are excluded instead of being filled in like they would be in the first row.

	% Significant
Fill Missing with Avg	21.49382%
Fill Missing with Group Avg	19.91856%

Table 1

The first Manhattan plot calculated is displayed by figure 1. It shows that most of the p-values are very high (manhat values below 20), but there are a lot of low p-values that require further inspection.

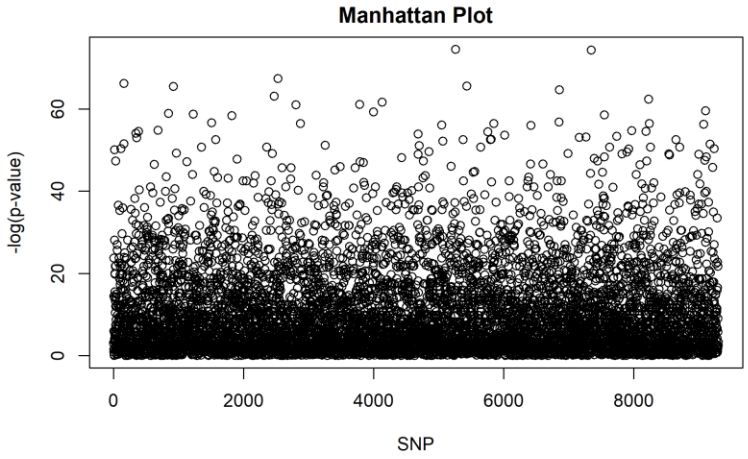


Figure 1

Figures 2 displays the same points as Figure 1 but with color differentiating the significant p-values for each cutoff. Blue points represent significant SNP's and red points represent the not-significant SNP's. The x-axis is in the original order of the SNP's given by the HapMap data set.

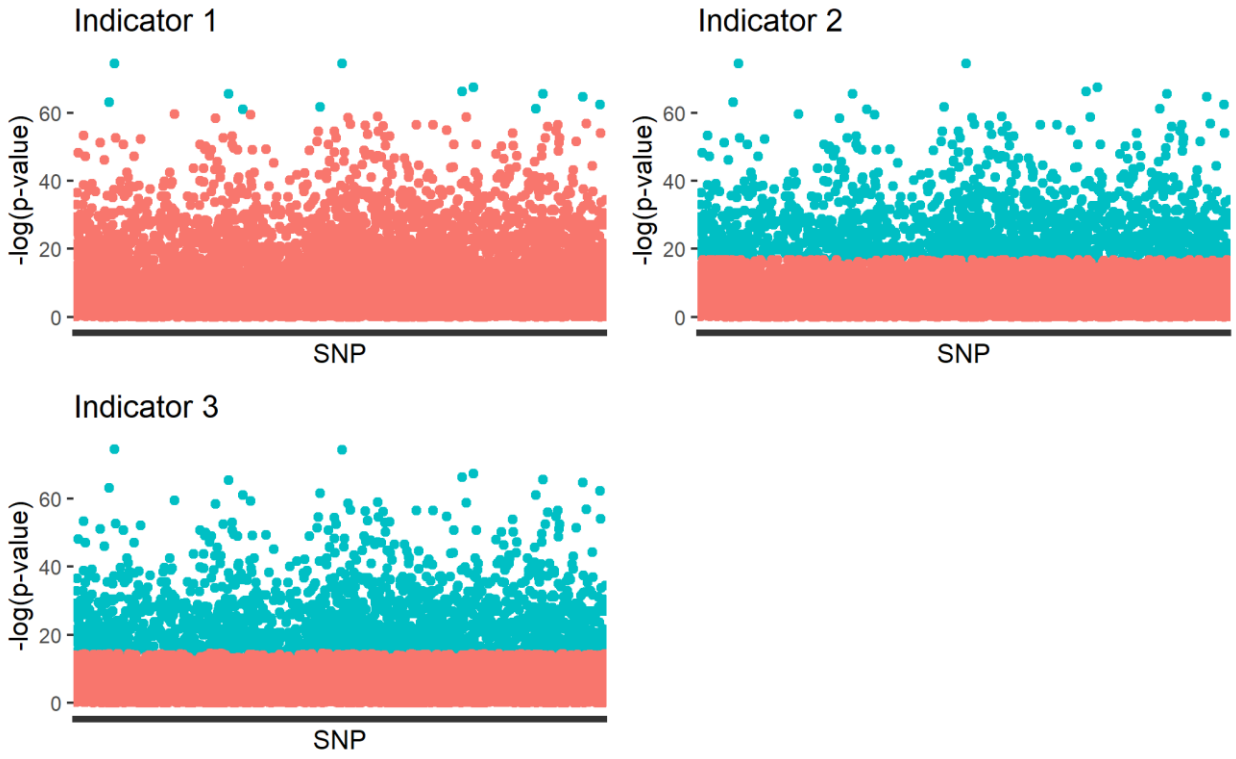


Figure 2

Figure 3 uses the same x-axis and shows the y-axis significant (y = 1) or not (y = 0) as a way to highlight grouping of significant p-values.

Figure 3

Table 2 shows how many significant SNP's were found. Highlighted is the percent of SNP's whose association with race is statistically significant for each of the significance cutoffs. For example, when the cutoff for significant SNP's is manhat values greater than 60, .15% of the SNP's are found to be significant. The second row shows the same results for the data where missing cells were filled with the group mean instead of with the overall mean.

	Indicator 1: Very Top (m > 60)	Indicator 2: Standard Cutoff (m > -log(5*10^(-8))	Indicator 3: Conservative Cutoff (m > -log(5*10^(-7))
% significant SNP's Method 1	.15% (12/7638)	19.55% (1493/7638)	23.96% (1830/7638)
% significant SNP's Method 2	2.32% (17/7313)	19.19% (1403/7313)	23.72% (1735/7313)

Table 2

Figure 4 shows a Manhattan plot using p-values calculated with the data where missingness was filled in with the mean of the race of the row of the missing cell.

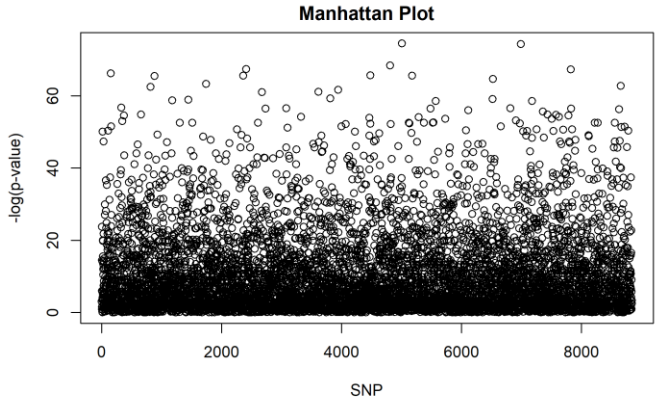


Figure 4

The top 10 most significant (lowest p-value) locations from the original analysis, listed from most to least significant, are:

rs10868791, rs2370893, rs6814827, rs6670842, rs7851392, rs13420968, rs9534610, rs10805068, rs9909962, and rs2034510.

The top 10 most significant locations using data that filled missing values with the group mean are:

rs10868791, rs2370893, rs368297, rs6814827, rs9909962, rs6670842, rs10504132, rs10805068, rs7851392, and rs13420968.

Discussion

Two assumptions are made when calculating missingness. One is that when a table is created of a SNP's genotypes, AA Aa aa is always the order the frequencies are presented. The second is that when two genotypes are present, they are both homozygous. The first assumption is reasonable based on the way R constructs tables. The latter is reasonable because the probability of all 120 subjects having any SNP be all Aa is extremely low.

More than 20% of the SNP's show significant evidence of there being no Hardy-Weinberg Equilibrium (and roughly 20% are significant when missing values are filled with the group

average). Therefore, we conclude the population has not reached Hardy-Weinberg Equilibrium. This is unsurprising because of the makeup of the sample – half from Utah, US and half from Ibadan, Nigeria – results in challenges to all of the assumptions.

Random mating is not met because individuals in Utah are much more likely to mate with others in Utah than they are to mate with individuals in Ibadan. Culture, language, and social status also impact human mating choice. For the second assumption, it is unclear how much migration is occurring in the 120 subjects and their ancestors, but it is reasonable that many of them could be first or second generation residents of the sampled area. Particularly in the US, moving across state lines is not uncommon. For the assumption that no mutations are occurring, we cannot guarantee that is true in a sample of the general population. Mutation is always possible in a random environment. The same is true for the assumption that no genetic drift occurs. Finally, the natural selection process in humans is challenged by modern medicine. These issues with the five assumptions are likely what is preventing the population from reaching Hardy-Weinberg Equilibrium.

Figure 1 shows that while most SNP's are not at all significant, there are a few that are extremely significant and a lot that are moderately significant. The plot of indicator 1 in Figure 2 helps us see the 12 most significant SNP's that had manhat values above an arbitrary number of 60. Significance indicator 1 was defined based on my understanding of the data and desire to explore trends. Indicator 1 was even created after the Manhattan Plot, so is not a limiting reflection of SNP impact on race. The plots of indicators 2 and 3 in Figure 2 are very similar with slightly more significant SNP's (blue points) in indicator 3, confirming that our calculation of indicator 3 was done properly since that is the slightly more conservative cutoff.

Figure 3 produced minimal helpful information. This was done with the assumption that the original SNP order is associated to SNP location on the DNA strand, but that is not specified anywhere that I could find. Indicators 2 and 3 resulted in too many significant SNP's to identify a pattern. However, the significant SNP's using indicator 1, the SNP's with the most extreme significance, seem to be grouped into twos and threes. This could be random chance, or it could

be that one SNP from each group is associated with race while the others are simply associated with the first SNP because of physical proximity on a chromosome.

Table 2 shows that based on the arbitrary cutoffs chosen in this study for indicator 1, 0.15% of all SNP's looked at this this study had a manhat value of more than 60 (indicator 1). This indicator was helpful for identifying the most significant SNP's and setting up their calculations was helpful in understanding the dataset. It was through that process that that tall_pval matrix was confirmed to hold empty rows and also that only locations with multiple genotypes should be included in the denominator of the calculation of percent of SNP's that are significant and not all 9307 locations of the original dataset.

Indicator 2 has the benefit of limiting false positives and has been shown to be effective for data that is not overwhelmingly large, which makes it appropriate for this data (Zhongsheng, 2021). Table 2 shows that based on this cutoff, 19.55% of SNP's are significantly associated with race. Indicator 3 is calculated with a more conservative cutoff which decreases the number of false negatives at the risk of increasing the number of false positives. Using this cutoff, 23.96% of SNP's are significantly associated with race.

In the methods section, we clarified that indicator 2 is the primary cutoff. With 19.55% of SNP's being significantly associated with race, we can conclude that a very large portion of SNP's are associated with race. Race in this analysis is either Ibadan or Utahn. It is assumed that these populations represent Black and Caucasian people respectively, but that is not explicitly stated. Ultimately, a large part of human DNA is associated with race in some manner, but it is unclear how much effect each SNP has on race when all other SNP's are present. There is no SNP that stands out as exceptionally more associated with race than any other, although that may change if the sample size was increased.

Both 19.55% and 23.96% are large percentages. Since the p-values were calculated using single variable logistic regression, it is possible that just a handful of the top 10 most significant variables determine race and that the rest of the significant variables are associated with those limited variables. The next step to exploring that is to build a multi-variable logistic regression model using forward selection. Due to time and processing power limitations, that model was

not within the scope of this paper. To proceed, compare a model with just rs10868791, the most significant SNP, to models with rs10868791 and each remaining SNP one at a time.

Our sensitivity analysis showed that while a few of the most significant variables got even more significant (15 SNP's with manhat values above 60 increased to 17), most SNP's decreased in significance (both percentage and number of p-values below -log(5*10^(-8)) and -log(5*10^(-7)) decreased). In both missingness-methods, the number of significant SNP's is higher with indicator 3, confirming it acts as the conservative indicator.

The lists of top 10 most significant variables shows the addition of two SNP's previously not included in the top 10 (rs368297 and rs10504132) and a reordering of some others (rs9909962 became more significant while rs7851392 and rs13420968 became less significant relative to the other SNP's). This volatility shows that missing values and how they are handled have an impact on conclusions about the association of SNP's and race. Other methods of filling missing values should be explored, including filling missing values with a probability distribution that weights genotypes by observed frequency within race.

In conclusion, major assumptions are broken that prevent this population from reaching Hardy-Weinberg Equilibrium. Simple logistic regression was used to find that approximately 19.55% of SNP's are significantly associated with race, meaning that race is not determined by just one or two genes. Rather, a multitude of genes collectively contribute to popular phenotypes that society classified as races. Future research may include using forward regression to build a multi-variable logistic regression model and focusing on the reasons for associations between SNP's.

The HapMap Project was founded with the intention of improving both healthcare and the understanding of human genes. Identifying so many SNP's that are associated with race has far reaching implications for other Genome-Wide Association Studies that may have overlooked race as a potential confounding variable. When treating people of different races, being aware of prominent genotypes and the potential risk factors associated with them is an important part of patient care. This analysis is a reminder to researchers and clinicians that race is genetically complex and is not defined by any one single SNP.

References

"About the International HapMap Project." *Genome.gov*, National Human Genome Research Institute, 4 June 2012, https://www.genome.gov/11511175/about-the-international-hapmap-project-fact-sheet#al-1.

Graffelman, Jan. "The Hardyweinberg Package - Uaem.mx." *The HardyWeinberg Package*, Department of Statistics and Operations Research Universitat Polit`Ecnica De Catalunya, Nov. 2009, http://www2.uaem.mx/r-mirror/web/packages/HardyWeinberg/vignettes/HardyWeinberg.pdf.

"Haplotype / Haplotypes." *Nature News*, Nature Publishing Group, 2014, https://www.nature.com/scitable/definition/haplotype-haplotypes-142/#:~:text=In%20addition%2C%20the%20term%20%22haplotype,the%20DNA%20sequence%20among%20individuals.

Hardy-Weinberg Equilibrium, Northern Arizona University, https://www2.nau.edu/lrm22/lessons/hwe/hwe.htm.

Zhongsheng Chen, Michael Boehnke, Xiaoquan Wen, Bhramar Mukherjee, Revisiting the genome-wide significance threshold for common variant GWAS, *G3 Genes|Genomes|Genetics*, Volume 11, Issue 2, February 2021, jkaa056, https://doi.org/10.1093/g3journal/jkaa056

GWAS Project

Jenny Eberling

2022-05-05

Code

Missingness

To fill in the missing SNP locations in the HapMap, the most frequently occuring genotype for each SNP is used. To attain the mode information, a loop finds the mode of each column (SNP). Within the first loop is a secondary loop that runs through every row within the column. If a cell has the value NA - it is missing - then the value is replaced with the mode of the column.

Missing Values

Lets look at data and do "quality control."

```
summary(HapMap$group)
```

```
## CEU YRI
##  60  60
```

```
# how many na's?
table(is.na(HapMap)) # 49002/(49002+1067838) = .04387
```

```
##
##   FALSE    TRUE
## 1067838   49002
```

```
# what to do with na's? Every observation has something missing - problem!

# make them whatever the majority genotype is
HM0 <- as.matrix(HapMap)
fullmat<-matrix(NA,nrow=120,ncol=9307) # create an empty matrix where we will
fill empty values


for (j in 3:9307){
  highfreq <- names(which.max(table(HM0[,j]))) # find mode of each column

  for (i in 1:120){
    fullmat[i,j] <- ifelse(is.na(HM0[i,j]) == TRUE, highfreq, HM0[i,j]) # if
empty cell, use mode
  }
```

```
}

# did it work?
table(is.na(fullmat)) # yes, only first two columns (id and group) are
missing

##
##   FALSE    TRUE
## 1116600     240

# make it back into a dataframe with id and group
HM_full <- as.data.frame(fullmat) # make df
names(HM_full) <- names(HapMap)   # put col names back
HM_full$id <- HapMap$id           # fill id variable
HM_full$group <- HapMap$group     # fill group (race) variable
```

Check for Hardy-Weinberg equilibrium

If HWE is acceptably met for 95% of SNP's, I will say HWE is met.

Assumptions: 1. No selection
2. No mutation
3. No migration
4. Large population
5. Random mating

These are not acceptably met.

```
#table(HM_full[,3])
# table(HM_full[,4])
n=120
hwe <- matrix(NA, nrow=4, ncol=9307)

for (j in 3:9307){
  if (length(names(table(HM_full[,j]))) == 3) {
  box <- as.data.frame(table(HM_full[,j], dnn = list("gene")))
  boxmat <- as.matrix(box)
  AA <- as.numeric(boxmat[1,2])
  Aa <- as.numeric(boxmat[2,2])
  aa <- as.numeric(boxmat[3,2])
  boxsep <- separate(box, gene, into = c("allele1", "allele2"), sep = 1)
#each on their own
  long <- pivot_longer(boxsep,c("allele1","allele2"),"allele") # stacks so
alleles separate
  summ <- long %>% group_by(value) %>% summarize(total_freq = sum(Freq))

  # Assumes order of table is AA Aa aa
  p = as.numeric(summ[1,2]/240) # frequency of A
  q = as.numeric(summ[2,2]/240) # frequency of a
```

```
  # do a test to see if pn and qn and pqn are equal to the table numbers
  hwe[1,j] <- HWChisq(c(AA,Aa,aa), verbose = FALSE)$pval
  hwe[2,j] <- p
  hwe[3,j] <- q
} else if (length(names(table(HM_full[,j]))) == 2) {
  # first, I want df to store my two alleles
  box <- as.data.frame(table(HM_full[,j], dnn = list("gene")))
  boxmat <- as.matrix(box)
  boxsep <- separate(box, gene, into = c("allele1", "allele2"), sep = 1)
  long <- pivot_longer(boxsep,c("allele1","allele2"),"allele") # stacks so
alleles separate
  # below code makes sure AA and Aa are correctly assigned
  if (boxsep[1,1]==boxsep[1,2]){
      AA <- as.numeric(boxmat[1,2])
      Aa <- as.numeric(boxmat[2,2])
      aa <- 0
  } else{
      AA <- as.numeric(boxmat[2,2])
      Aa <- as.numeric(boxmat[1,2])
      aa <- 0
  }
  summ <- long %>% group_by(value) %>% summarize(total_freq = sum(Freq))

  # I'm going to assume there are no SNP's with two homologous alleles
  p = as.numeric(summ[1,2]/240)
  q = as.numeric(summ[2,2]/240)

  hwe[1,j] <- HWChisq(c(AA,Aa,aa),verbose = FALSE)$pval
  hwe[2,j] <- p
  hwe[3,j] <- q
} else if (length(names(table(HM_full[,j]))) == 1) { # assumes no SNP with
all Aa genotype
  box <- as.data.frame(table(HM_full[,j], dnn = list("gene")))
  boxmat <- as.matrix(box)
  AA <- as.numeric(boxmat[1,2])
  Aa <- 0
  aa <- 0

  p = as.numeric(2*table(HM_full[,j])[1]/240)
  q = 0

  hwe[1,j] <-HWChisq(c(AA,Aa,aa),verbose = FALSE)$pval
  hwe[2,j] <- p
  hwe[3,j] <- q
}
}
```

```
hwe <- as.data.frame(hwe)
names(hwe) <- names(HM_full)
row.names(hwe) <- c("hwe_pval","p","q","h")
hwe <- hwe %>% select(-c("id","group"))
hwe<-hwe[1:3,1:9305]
```

To determine HWE, I determining if more than 5% of p-vaues are significant. 21.49% of p-values are significant, so HWE is not reached in this population.

```
hwe_tall <- as.data.frame(t(hwe))
hwe_tall$signif <- ifelse(hwe_tall$hwe_pval < .05, 1, 0)
table(hwe_tall$signif) %>% prop.table()

##
##         0         1
## 0.7850618 0.2149382
```

Logistic regression

I performed simple logistic regression on race (group) with each variable and saved the p-values.

```
# don't run on variables with no variability
varlist <- names(HM_full[,3:9307]) # get a list without id and group
pval_list <- matrix(NA,nrow=2,ncol=9305) # every variable will have a p-val
base_model <- glm(group ~ 1, family = 'binomial', data=HM_full) ##"null" mod
(need this to calc p-val)



##################
# loop regression
##################
for (i in seq_along(varlist)){
  # First, create null model needed to calculate p-val.
  # It needs to be done like this so it's comparable to the models made in
the loop
  base_blueprint <- as.formula(sprintf("group ~ 1"))
  base_model <- glm(formula = base_blueprint, family = binomial, data =
HM_full)


  # I only want to run when there are multiple levels of a genotype
  if (length(table(HM_full[,i+2])) != 1 ){
    # code_blueprint lets me change the x variable in my glm model
    code_blueprint <- as.formula(sprintf("group ~ %s", varlist[i]))
    # create glmmodel with one variable each time
    glm_model <- glm(formula = code_blueprint, family = binomial, data =
HM_full)
```

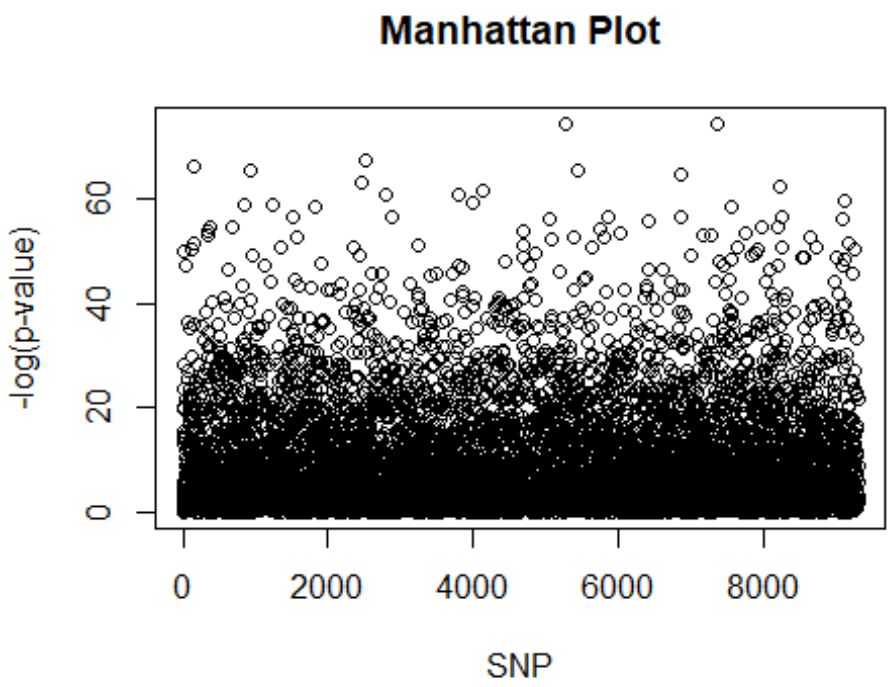
```
    # Find p-value by comparing each model to the null model. I want 2nd
pval, the one from hm_full.
    pval_list[1,i] <- varlist[i] # first row is col name
    pval_list[2,i] <- anova(base_model, glm_model, test =
'Chisq')$"Pr(>Chi)"[2]
  }
}
```

Manhattan Plot

Using the p-values attained above, I made a Manhattan plot.

```
### Setup ###
# make matrix a data frame
pval_data <- data.frame(pval_list)
tall_pval <- data.frame(t(pval_data))
tall_pval$manhat <- -1*log(as.numeric(tall_pval$X2))


# pch = 19 to fill circles
plot(tall_pval$manhat, xlab = "SNP",ylab = "-log(p-value)",main = "Manhattan
Plot", pch = 21)
```



So which SNP's are important? I looked at three different cutoffs. 1. p-value is less than 8.756511e-27, chosen arbitrarily at first glance of the original Manhattan plot
2. p-value is less than 5*10^(-8), an industry standard cutoff
3. p-value is less than 5*10^(-7), a conservative industry standard cutoff

The top 10 most significant SNP's are rs10868791, rs2370893, rs6814827, rs6670842, rs7851392, rs13420968, rs9534610, rs10805068, rs9909962, and rs2034510.

```
# Above 60
tall_pval <- tall_pval %>% mutate(ind1 = as.factor(ifelse(manhat > 60,1,0)))
# rs10868791

# typical cutoff of 5 × 10^-8 - there are 1493 significant SNPs
tall_pval <- tall_pval %>% mutate(ind2 = as.factor(ifelse(manhat > -
1*log(5*10^(-8)),1,0)))
tall_pval <- tall_pval %>% mutate(ind3 = as.factor(ifelse(manhat > -
1*log(5*10^(-7)),1,0)))

tall_pval <- tall_pval %>%
  mutate(original_order = 1:nrow(tall_pval)) %>% # save the original order of
SNP's
  arrange(-manhat) %>%
  mutate(order = 1:nrow(tall_pval))             # order p-values lowest to
highest



table(tall_pval$ind1)

##
##    0    1
## 7626   12

table(tall_pval$ind2)

##
##    0    1
## 6145 1493

table(tall_pval$ind3)

##
##    0    1
## 5808 1830

#tall_pval %>% filter(is.na(X1)==T)
```

Let's look at these cutoffs in a manhattan plot.

```
plot1 <- tall_pval %>% drop_na() %>%
  ggplot() +
  geom_point(aes(x=X1, y=manhat, col=ind1)) +
  labs(x = 'SNP',y = '-log(p-value)', title = 'Indicator 1') +
  theme(legend.position = 'none',axis.text.x=element_blank())

plot2 <- tall_pval %>% drop_na() %>%
```

```
  ggplot() +
  geom_point(aes(x=X1, y=manhat, col=ind2)) +
  labs(x = 'SNP',y = '-log(p-value)', title = 'Indicator 2') +
  theme(legend.position = 'none',axis.text.x=element_blank())

plot3 <- tall_pval %>% drop_na() %>%
  ggplot() +
  geom_point(aes(x=X1, y=manhat, col=ind3)) +
  labs(x = 'SNP',y = '-log(p-value)', title = 'Indicator 3') +
  theme(legend.position = 'none',axis.text.x=element_blank())

grid.arrange(plot1,plot2,plot3, ncol=2)
```

One thing that I notice is that there isn't apparent clumping of the significant SNP's. They appear evenly dispersed. When I look at significance against the original order of SNP's given to us, there is not one section where more SNP's are significant in predicting race than another spot.

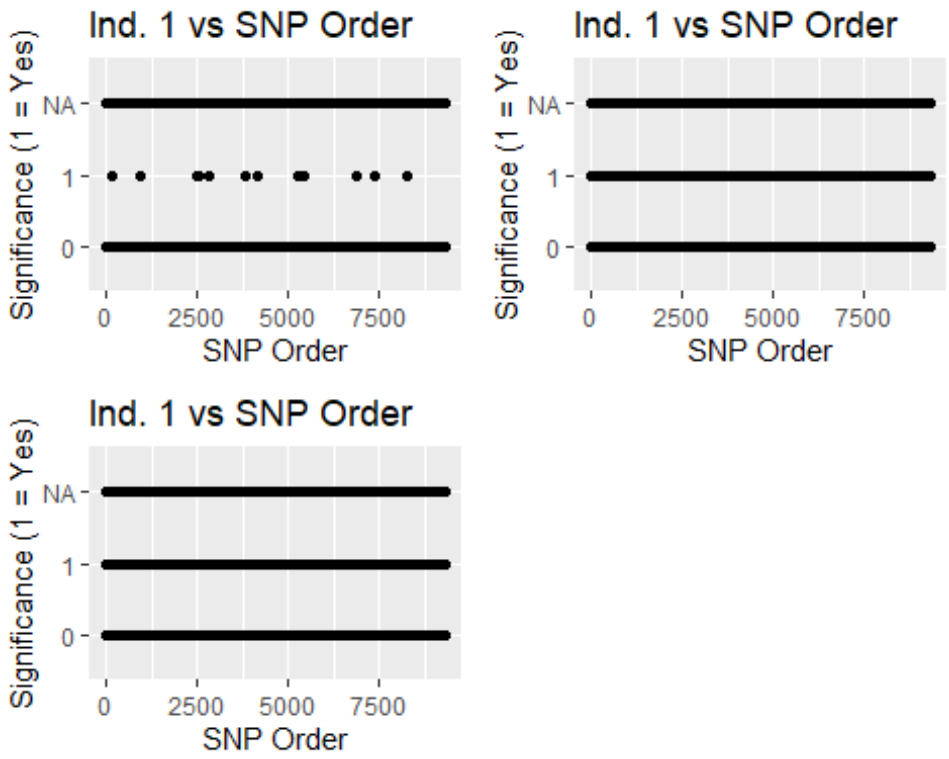
```
plot4 <- tall_pval %>% ggplot(aes(x=original_order, y=as.factor(ind1))) +
  geom_point() +
  labs(title = "Ind. 1 vs SNP Order", x = "SNP Order", y = "Significance (1 =
Yes)")

plot5 <- tall_pval %>% ggplot(aes(x=original_order, y=as.factor(ind2))) +
  geom_point() +
  labs(title = "Ind. 1 vs SNP Order", x = "SNP Order", y = "Significance (1 =
```

```
Yes)")

plot6 <- tall_pval %>% ggplot(aes(x=original_order, y=as.factor(ind3))) +
  geom_point() +
  labs(title = "Ind. 1 vs SNP Order", x = "SNP Order", y = "Significance (1 =
Yes)")


grid.arrange(plot4, plot5,plot6,ncol=2)
```



I would love to build a full logistic model using forward selection, but I do not have the computing power to do it efficiently.

Comparison

I then wanted to try a different method of filling in HapMap missing values and see if that affects the important variables.

```
# make them whatever the majority genotype is * by race *
#HM0 <- as.matrix(HapMap)[,-c(1,2)] # drop id and group
HM0 <- as.matrix(HapMapOriginal)
fullmatgroup<-matrix(NA,nrow=120,ncol=9307) # create an empty matrix where we
will fill empty values
empty_columns <- c() # columns empty for all of a group go in here

# if all 60 of a race is missing, make the whole column 0
for (j in 3:9307){
```

```
  missingceu <- c(is.na(HM0[1:60,j]))
  missingyri <- c(is.na(HM0[61:120,j]))
  allmissing <- c(rep(T,60))

  if (setequal(missingceu,allmissing)==T |
setequal(missingyri,allmissing)==T){
    empty_columns <- c(empty_columns, j)
  }
}


HMCEU <- as.matrix(HapMapOriginal %>% filter(group=="CEU"))
HMYRI <- as.matrix(HapMapOriginal %>% filter(group=="YRI"))


# One for CEU values
for (j in 3:9307){
  if ((!(j %in% empty_columns))==T){
  highfreqCEU <- names(which.max(table(HMCEU[,j])))
  for (i in 1:60){
    fullmatgroup[i,j] <- ifelse(is.na(HM0[i,j]) == TRUE, highfreqCEU,
HM0[i,j])
  }
  }
}

# Add on the YRI values
for (j in 3:9307){
  if ((!(j %in% empty_columns))==T){
  highfreqYRI <- names(which.max(table(HMYRI[,j])))
  for (i in 61:120){
    fullmatgroup[i,j] <- ifelse(is.na(HM0[i,j]) == TRUE, highfreqYRI,
HM0[i,j])
  }
  }
}

n_distinct(empty_columns)*120+240 # empty columns plus id and group are empty

## [1] 55920

table(is.na(fullmatgroup)) # it matches! that means it's filled in correctly

##
##   FALSE    TRUE
## 1060920   55920

# make it back into a dataframe with id and group
HM_fullgroup <- as.data.frame(fullmatgroup) # make df
names(HM_fullgroup) <- names(HapMap)   # put col names back
```

```
HM_fullgroup$id <- HapMap$id           # fill id variable
HM_fullgroup$group <- HapMap$group     # fill group (race) variable

HM_fullgroup_t <- t(HM_fullgroup)
HM_fullgroup_t <- na.omit(HM_fullgroup_t)
HM_fullgroup <- as.data.frame(t(HM_fullgroup_t))

# These should match
ncol(fullmatgroup) - n_distinct(empty_columns)

## [1] 8843

ncol(HM_fullgroup)

## [1] 8843
```

Is this population in HWE?

```
n=120
hwegroup <- matrix(NA, nrow=4, ncol=8843)
ncol(as.data.frame(HM_fullgroup))

## [1] 8843

for (j in 3:8843){
    if (length(names(table(HM_fullgroup[,j]))) == 3) {
    box <- as.data.frame(table(HM_fullgroup[,j], dnn = list("gene")))
    boxmat <- as.matrix(box)
    AA <- as.numeric(boxmat[1,2])
    Aa <- as.numeric(boxmat[2,2])
    aa <- as.numeric(boxmat[3,2])
    boxsep <- separate(box, gene, into = c("allele1", "allele2"), sep = 1)
    long <- pivot_longer(boxsep,c("allele1","allele2"),"allele")
    summ <- long %>% group_by(value) %>% summarize(total_freq = sum(Freq))

    # Assumes order of table is AA Aa aa
    p = as.numeric(summ[1,2]/240) # frequency of A
    q = as.numeric(summ[2,2]/240) # frequency of a

    hwegroup[1,j] <- HWChisq(c(AA,Aa,aa), verbose = FALSE)$pval
    hwegroup[2,j] <- p
    hwegroup[3,j] <- q
  } else if (length(names(table(HM_fullgroup[,j]))) == 2) {
    # first, I want df to store my two alleles
    box <- as.data.frame(table(HM_fullgroup[,j], dnn = list("gene")))
    boxmat <- as.matrix(box)
    boxsep <- separate(box, gene, into = c("allele1", "allele2"), sep = 1)
    long <- pivot_longer(boxsep,c("allele1","allele2"),"allele") # stacks so
alleles separate
    # below code makes sure AA and Aa are correctly assigned
    if (boxsep[1,1]==boxsep[1,2]){
```

```
        AA <- as.numeric(boxmat[1,2])
        Aa <- as.numeric(boxmat[2,2])
        aa <- 0
    } else{
        AA <- as.numeric(boxmat[2,2])
        Aa <- as.numeric(boxmat[1,2])
        aa <- 0
    }
    summ <- long %>% group_by(value) %>% summarize(total_freq = sum(Freq))
    # I'm going to assume there are no SNP's with two homologous alleles
    p = as.numeric(summ[1,2]/240)
    q = as.numeric(summ[2,2]/240)

    hwegroup[1,j] <- HWChisq(c(AA,Aa,aa), verbose = FALSE)$pval
    hwegroup[2,j] <- p
    hwegroup[3,j] <- q
  } else if (length(names(table(HM_fullgroup[,j]))) == 1) { # assumes no SNP
with all Aa
    box <- as.data.frame(table(HM_fullgroup[,j], dnn = list("gene")))
    boxmat <- as.matrix(box)
    AA <- as.numeric(boxmat[1,2])
    Aa <- 0
    aa <- 0

    p = as.numeric(2*table(HM_fullgroup[,j])[1]/240)
    q = 0

    hwegroup[1,j] <-HWChisq(c(AA,Aa,aa), verbose = FALSE)$pval
    hwegroup[2,j] <- p
    hwegroup[3,j] <- q
}
}


hwegroup <- as.data.frame(hwegroup)
names(hwegroup) <- names(HM_fullgroup)
row.names(hwegroup) <- c("hwe_pval","p","q","h")
hwegroup <- hwegroup %>% select(-c("id","group"))
hwegroup<-hwegroup[1:3,1:8841]
```

Still more than 5% are significant. No HWE.

```
hwe_tallgroup <- as.data.frame(t(hwegroup))
hwe_tallgroup$signif <- ifelse(hwe_tallgroup$hwe_pval < .05, 1, 0)
table(hwe_tallgroup$signif) %>% prop.table()

##
##         0         1
## 0.8008144 0.1991856
```

Does regression look different with this data that's filled in differently?

```
# don't run on variables with no variability
varlist <- names(HM_fullgroup[,3:8843]) # get a list without id and group
pval_list <- matrix(NA,nrow=2,ncol=8841) # every variable will have a p-val
HM_fullgroup$group <- as.factor(HM_fullgroup$group)
base_model <- glm(group ~ 1, family = 'binomial', data=HM_fullgroup) ##"null"
mod (need this to calc p-val)




###################
# loop regression
###################
for (i in seq_along(varlist)){
  base_blueprint <- as.formula(sprintf("group ~ 1"))
  base_model <- glm(formula = base_blueprint, family = binomial, data =
HM_fullgroup)

  if (length(table(HM_fullgroup[,i+2])) != 1 ){
    code_blueprint <- as.formula(sprintf("group ~ %s", varlist[i]))
    glm_model <- glm(formula = code_blueprint, family = binomial, data =
HM_fullgroup)
    pval_list[1,i] <- varlist[i] # first row is col name
    pval_list[2,i] <- anova(base_model, glm_model, test =
'Chisq')$"Pr(>Chi)"[2]
  }
}
```

Let's now look at significance

```
### Setup ###
# make matrix a data frame
pval_datagroup <- data.frame(pval_list)
tall_pvalgroup <- data.frame(t(pval_datagroup))
tall_pvalgroup$manhat <- -1*log(as.numeric(tall_pvalgroup$X2))

# Above 60
tall_pvalgroup <- tall_pvalgroup %>% mutate(ind1 = as.factor(ifelse(manhat >
60,1,0)))  # rs10868791

# typical cutoff of 5 × 10^-8 - there are 1493 significant SNPs
tall_pvalgroup <- tall_pvalgroup %>%
  mutate(ind2 = as.factor(ifelse(manhat > -1*log(5*10^(-8)),1,0)))
tall_pvalgroup <- tall_pvalgroup %>%
  mutate(ind3 = as.factor(ifelse(manhat > -1*log(5*10^(-7)),1,0)))


tall_pvalgroup <- tall_pvalgroup %>%
```

```
  mutate(original_order = 1:nrow(tall_pvalgroup)) %>% # save the original
order of SNP's
  arrange(-manhat) %>%
  mutate(order = 1:nrow(tall_pvalgroup))

table(tall_pvalgroup$ind1)

##
##    0    1
## 7296   17

table(tall_pvalgroup$ind2)

##
##    0    1
## 5910 1403

table(tall_pvalgroup$ind3)

##
##    0    1
## 5578 1735

plot(x=tall_pvalgroup$original_order,y=tall_pvalgroup$manhat, xlab =
"SNP",ylab = "-log(p-value)",main = "Manhattan Plot", pch = 21)
```