704 Assignment 1 - GWAS

Peter Shen

Sys.date()

Background Questions

1) The diseases of interested in WTCCC is bipolar disorder (BD), coronary artery disease (CAD), Crohn's disease (CD), hypertension (HT), rheumatoid arthritis (RA), type 1 diabetes (T1D), and type 2 diabetes (T2D).

E + G -> P -> D

- E: None examined
- G: 500K SNP Microarray Data
- P: Gene expression of the variant on causing phenotypic changes to things such as elevated blood pressure, blood glucose, or blood tryglycerides, etc.
- D: bipolar disorder (BD), coronary artery disease (CAD), Crohn's disease (CD), hypertension (HT), rheumatoid arthritis (RA), type 1 diabetes (T1D), and type 2 diabetes (T2D)

2)

- **BD**: Individuals have suffered one or more episodes of pathologically elevated mood. They were interviewed by psychiatrists and measured using a OPCRIT checklist and psychiatric diagnoses were assigned using Research Diagnostic Criteria.
- **CAD**: Hospital validated history of either myocardial infarction or coronary revascularization (coronary artery bypass surgery or percutaneous coronary angioplasty) before their 66th birthday.
- **CD**: Conventional endoscopic, radiological and histopathological criteria.
- **HT**: History of hypertension diagnosed before 60 yr of age, with confirmed blood pressure recordings corresponding to seated levels.150/100mmHg (if based on one reading), or the mean of 3 readings greater than 145/95mmHg
- **RA**: Caucasian ver the age of 18 yr and satisfied the 1987 American College of Rheumatology Criteria for RA127 modified for genetic studie
- **T1D**: An age of diagnosis below 17 yr and have insulin dependence since diagnosis
- **T2D**: Based on either current prescribed treatment with sulphonylureas, biguanides, other oral agents and/or insulin or, in the case of individuals treated with diet alone, historical or contemporary laboratory evidence of hyperglycaemia.

3) Data collection was performed on 500k Affymetrix SNP Microarray Chip and then imputed against 2,193,483 HapMap SNPs not on the Affymetrix chip. The SNP chip uses DNA probes that are perfect complements to different SNP variant. The sample DNA hybridizes to either of the SNP probes, and they can be called using optical scans, which are then mapped to the SNP mapping of the chip to generate the matrix of the different SNP variants. Because these chips only have ~500k SNPs, the dataset is expanded through HapMap imputations, which work through via linkage disequilibrium (LD).

4) `$ wc -l *tped`

500568 total SNPs were assayed for each individual.

5)

```
df5 <- data.frame("CC"=c(270, 436), "CT"=c(957,1398), "TT"=c(771, 1170))
df5.additive <- df5 %>% mutate(C=CC*2+CT) %>% mutate("T"=CT+TT*2) %>% select("C", "T")
```

A)

```
df5.freq <- df5.additive %>%
  mutate(sum=C+`T`) %>%
  transmute(C=C/sum, `T`=`T`/sum)
df5.freq
```

```
##           C         T
## 1 0.3746246 0.6253754
## 2 0.3778296 0.6221704
```

B)

```
chisq.test(df5.additive, correct = F)
```

```
##
##  Pearson's Chi-squared test
##
## data:  df5.additive
## X-squared = 0.105, df = 1, p-value = 0.7459
```

With a p-value of 0.746, we do not have strong enough evidence to conclude there is an association between rsGOINGALLIN and bipolar disorder.

C)

```
df5.control <- df5[2,]
N <- sum(df5.control)
df5.hw <- df5.control %>% transmute(p=(2*CC+CT)/(2*CC+2*CT+2*TT)) %>% mutate(q=1-p) %>% transmute(CC=N*

df5.hw <-  bind_rows(df5.control, df5.hw)
df5.hw.q <- sum((df5.hw[1,]-df5.hw[2,])^2 / df5.hw[2,])
pchisq(df5.hw.q, df=1, lower.tail = F)
```

```
## [1] 0.5782178
```

With a p-value of less than 0.578, we do not have strong enough evidence to refute the null hypothesis that the control and disease allele frequencies are in Hardy-Weinberg equilibrium.

Execution of GWAS

1) Executing GWAS for small set of SNPs

Import sample data for chromosome 22.

```
control.58c <- read.delim('sampledata/Affx_gt_58C_Chiamo_22.tped.gz', sep="\t", header=F, stringsAsFact
control.nbs <- read.delim('sampledata/Affx_gt_NBS_Chiamo_22.tped.gz', sep="\t", header=F, stringsAsFact
disease <- read.delim('sampledata/Affx_gt_T2D_Chiamo_22.tped.gz', sep="\t", header=F)
snps <- read.delim('sampledata/snps_22', header=F)
chrom <- 22
```

Combine the two control samples

```
control <- control.58c %>% bind_cols(control.nbs[,5:ncol(control.nbs)])
```

Sample with 1 SNP

```
# Scaling done on O2

GWA <- function(csnp, dsnp) {
  # build contingency table
  snpTable <- table(t(csnp[5:ncol(csnp)]), useNA="no") %>% bind_rows(table(t(dsnp[5:ncol(dsnp)]), useNA
  snpTable[is.na(snpTable)] <- 0

  # check for edge cases with alleles
  if (ncol(snpTable)==1){
    # if there is only 1 SNP, there's not going to be any contributional effect from this snp so just s
    return(NA)
  } else if(ncol(snpTable)==2) {
    # if there are 2 SNPS found in the samples, then we just add an empty count for the third possible
    snpTable <- bind_cols(snpTable, data.frame(`N N`=c(0,0)))
  }

  snpTable.colNames <- unlist(strsplit(colnames(snpTable[,2]), " "))
  snpTable.additive <- snpTable %>% transmute(a1=.[[1]]*2+.[[2]], a2=.[[2]]+.[[3]]*2)
  colnames(snpTable.additive) <- snpTable.colNames

  # determine major and minor alleles
  majorAllele <- max.col(snpTable.additive[1,])
  minorAllele <- colnames(snpTable.additive[1,-majorAllele])
  majorAllele <- colnames(snpTable.additive[1,majorAllele])

  return(c (`Chrom`=as.numeric(csnp[1]),
            `WTCCC`=as.character(csnp[[2]]),
            `MinorAllele`=minorAllele,
            `MajorAllele`=majorAllele,
            `ControlMajorAlleleCount`=snpTable.additive[[1,majorAllele]],
            `ControlMinorAlleleCount`=snpTable.additive[[1,minorAllele]],
            `DiseaseMajorAlleleCount`=snpTable.additive[[2,majorAllele]],
            `DiseaseMinorAlleleCount`=snpTable.additive[[2,minorAllele]],
            `AACount`=snpTable[[1,1]],
            `ABCount`=snpTable[[1,2]],
            `BBCount`=snpTable[[1,3]]) )

}

# QC for files
print(paste("Performing QC for chromosome", chrom))
control.misread <- which(control$V1==chrom)
control <- control[control.misread,]
disease <- disease[control.misread,]

disease.misread <- which(disease$V1==chrom)
control <- control[disease.misread,]
disease <- disease[disease.misread,]

print(paste("Performing GWA for chromosome", chrom))
gwaResult <- do.call(rbind, lapply(seq_along(1:1), function(i){GWA(control[i,], disease[i,])}))
gwaResult <- data.frame(gwaResult, stringsAsFactors = F)

print("Finished GWAS, writing out intermediate data")
# write.table(gwaResult, file=paste0("./imm/gwa", chrom, ".csv"), sep="\t")
```

```r
# Remove any rows with NA values
print("Remove rows with NA")
gwaResult <- gwaResult[complete.cases(gwaResult),]

# Class conversions
print("Class conversion")
gwaResult <- gwaResult %>% mutate(Chrom=as.numeric(Chrom),
                                  ControlMajorAlleleCount=as.numeric(ControlMajorAlleleCount),
                                  ControlMinorAlleleCount=as.numeric(ControlMinorAlleleCount),
                                  DiseaseMajorAlleleCount=as.numeric(DiseaseMajorAlleleCount),
                                  DiseaseMinorAlleleCount=as.numeric(DiseaseMinorAlleleCount),
                                  AACount=as.numeric(AACount),
                                  ABCount=as.numeric(ABCount),
                                  BBCount=as.numeric(BBCount))

# get RSID
print("Get RSID")
snps <- snps %>% select(V2, V4, V5) %>% transmute(WTCCC=V4, rsid=V5, pos=V2)
gwaResult <- gwaResult %>% left_join(snps, by='WTCCC')

print("Frequency Calculations")
gwaResult <- gwaResult %>%
  mutate(ControlMinAlleleFreq=ControlMinorAlleleCount/(ControlMinorAlleleCount+ControlMajorAlleleCount)
  mutate(DiseaseMinAlleleFreq=DiseaseMinorAlleleCount/(DiseaseMinorAlleleCount+DiseaseMajorAlleleCount)

print("Calculate OR Ratios")
gwaResult <- gwaResult %>% mutate(OR=(DiseaseMajorAlleleCount*ControlMinorAlleleCount)/(DiseaseMinorAlle

print("generate HW p and q values")
gwaResult <- gwaResult %>%
  mutate(p=(2*AACount+ABCount)/(2*AACount+2*ABCount+2*BBCount), total = AACount + ABCount + BBCount) %>%


# Chi-sq tests for SNP and HW
print("Chi-sq Test for SNPs")
gwaResult$PValue <- apply(gwaResult, 1, function(x) {
  chisq.test(matrix(
      as.numeric(c(x['ControlMajorAlleleCount'], x['ControlMinorAlleleCount'],
                   x['DiseaseMajorAlleleCount'], x['DiseaseMinorAlleleCount'])),
      nrow=2, ncol=2, byrow = T), correct = F)$p.value
  })

# HW chi-sq test
print("Chi-sq Test for HW")
gwaResult$HWPValue <- apply(gwaResult, 1, function(x) {
   m <-  matrix(
      as.numeric(c(x['AACount'], x['ABCount'], x['BBCount'],
                   x['HWAA'], x['HWAB'], x['HWBB'])),
      nrow=2, ncol=3, byrow = T)
   hw.test.q <- sum((m[1,]-m[2,])^2 / m[2,])
   return(pchisq(hw.test.q, df=1, lower.tail = F))
  })
```

```r
# concatenate to the results we want
print("Select results table")
gwaResult <- gwaResult %>% select(Chrom, rsid, pos, MinorAllele, MajorAllele, DiseaseMinAlleleFreq, Con

print("Write out results to file")
# write.table(gwaResult, file=paste0("./results/gwa", chrom, ".csv"), sep="\t")
```

Import the results of GWAS for T2D

```r
gwaResult <- read.delim('./results/gwa_results_T2D.csv', sep="\t", header=T)
```

1 A)

Filter out SNPs

```r
print(paste("SNPs before filter: ", nrow(gwaResult)))
```

```
## [1] "SNPs before filter:  481844"
```

```r
# Some SNPs have pvalues of 0
gwaResult <- gwaResult %>%  filter(PValue > 0 & HWPValue > 0)

# SNPs that deviate from Hardy-Weinberg because these are most likely due to genotyping errors
# and SNPs that have MAF < 0.01
gwaResult <- gwaResult %>%  filter(HWPValue >= 0.05 & ControlMinAlleleFreq >= 0.01 & DiseaseMinAlleleFre
print(paste("SNPs after filter: ", nrow(gwaResult)))
```

```
## [1] "SNPs after filter:  375974"
```

Calculate Bonferroni Threshold

```r
bf.pvalue <- 0.05/nrow(gwaResult)
print(paste("Bonferonni threshold: ", bf.pvalue))
```

```
## [1] "Bonferonni threshold:  1.32987919377404e-07"
```

1 B)

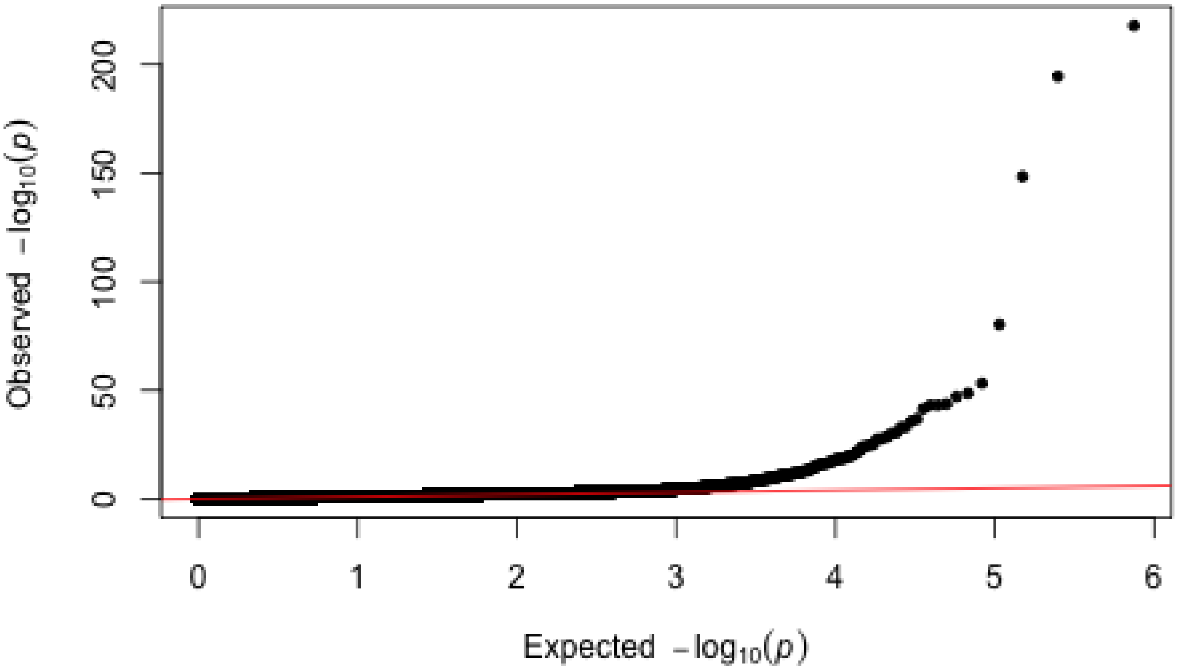
Write out the file to `T2D.csv`

```r
write.table(gwaResult, file="T2Ds.csv", sep="\t")
```

2 A)

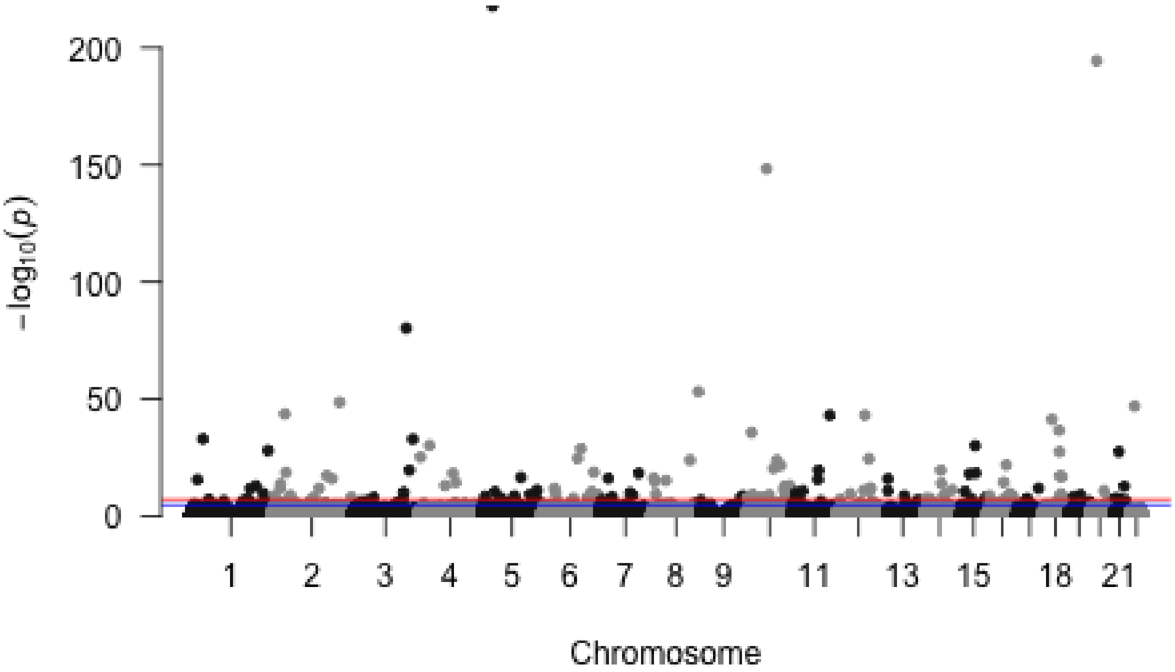
QQPlot of PValues

```r
gwaResult$logPvalue <- -log10(gwaResult$PValue)
# ggplot(gwaResult) + geom_qq(mapping=aes(sample=logPvalue), distribution = qunif) + geom_abline(interc
qqman::qq(gwaResult$PValue)
```



Manhattan Plot of PValues

```
manhattan.data <- gwaResult %>% select(SNP=rsid, CHR=Chrom, P=PValue, BP=pos)
manhattan(manhattan.data)
```



2 B)

SNPs exceeding Bonferonni threshold for T2D

```
gwaResult %>% filter(PValue <= bf.pvalue) %>% summarise(N=nrow(.), min=min(PValue), max = max(PValue), s
```

```
##     N          min          max           sd
## 1 159 1.198274e-218 1.279258e-07 2.813357e-08
```

2 C)

Some of the possible introductions of bias into the associations could be due to population effects – the effect of population stratification where individuals from different lineages have sufficiently different linkage disequilibrium profiles that is picked up by the association test and lead to significance of association. One could control for this phenonenon by either recalibrating the p-values based on genome inflation factor or control for it using PCA approaches such as used in EIGENSTRAT.

Another source of bias also due to LD is that the SNPs that was found to be associative to disease may not be the causal SNP since it could just be in LD with the causal SNP.

One source of bias is that if an SNP has a relatively small MAF (eg. < 5%) but would have a large effect size, we cannot reliably measure the effect of the SNPs due to lack of power to detect the SNP in the population.

2 D)

Investigating SNP `rs4506565`

```
snp.4506565 <- gwaResult[which(gwaResult$rsid=='rs4506565'),]
c(rsid=as.character(snp.4506565$rsid), chr=snp.4506565$Chrom,  pvalue=snp.4506565$PValue, OR=snp.4506565
```

```
##                   rsid                    chr                 pvalue 
##            "rs4506565"                   "10" "1.23475267389861e-13" 
##                     OR            MajorAllele            MinorAllele 
##    "0.730104466625885"                    "A"                    "T" 
```

For this SNP, since the OR denotes comparison of major allele to the minor allele, the odds of getting T2D for each allele `A` is decreased by 0.73, or inversely the odds of getting T2D for each allele `T` is increased by 1.37.

This is a SNP within the Transcription factor 7-like 2 (TCF7L2) gene (1). This SNP has been shown to be associated with T2D in several published papers(2,3,4). The TCF7L2 protein is a transcription factor that acts in the Wnt signalling pathway, which is an actor in transcription regulation, which in part may disrupt the expression of GLP-1 and lead to increased risk for T2D (5)

1. https://www.snpedia.com/index.php/Rs4506565
2. https://www.ncbi.nlm.nih.gov/pubmed/16936217?dopt=Abstract
3. https://www.ncbi.nlm.nih.gov/pubmed/16855264?dopt=Abstract
4. https://www.ncbi.nlm.nih.gov/pubmed/17206141?dopt=Abstract
5. https://www.ncbi.nlm.nih.gov/pubmed/18599616

3)

In additive models, each individual allele count linearly contributes to a phenotypic outcome. However, in a genotypic model, each genotype is associated with a phenotypic outcome instead.

To test the additive model using chi-squared association, as it is done in this assignment, we would need to count the sum of the individual alleles and execute a porportionality test (chi-square) between control and disease. To carry out a linear regression, we would need to reshape our data and assign numeric values (eg, 0 for `A` and 1 for `a`) to the count of alleles paired with their outcomes for each individual (disease vs. control), and then execute a linear regression model.

To test the genotypic model using porportionality test, we would do the same as before, but instead of executing the test on the sum of the individual alles, we cound sum genotypes of the population sample instead. To again carry out a linear regression, we would reshape our data such that each genotypic is its own variable in the model (eg. `X` for `AA`, `Y` for `AB`, and `Z` for `BB`) paired with their outcomes for the individual.

4)

In the background question, we are only looking at one candidate SNP that could cause association with disease. In GWAS, we are searching genome-wide for SNPs that are associated with disease. In GWAS, since p-values are uniformly distributed, we would have to correct for multiple hypothesis testing using Bonferonni threshold or FDR to control for false positives. GWAS gives greater power at detecting association compared to examining one single SNP.

5)

We could control for population stratification in the SNP array using PCA or adjusting for genome inflation factor.

6)

Calculating genome inflation factor

```
x2.values <- qchisq(1-gwaResult$PValue, 1) # other tail
x2.median.obs <- median(x2.values)
genome.IF <- x2.median.obs / 0.455
print(genome.IF)
```

```
## [1] 1.112138
```

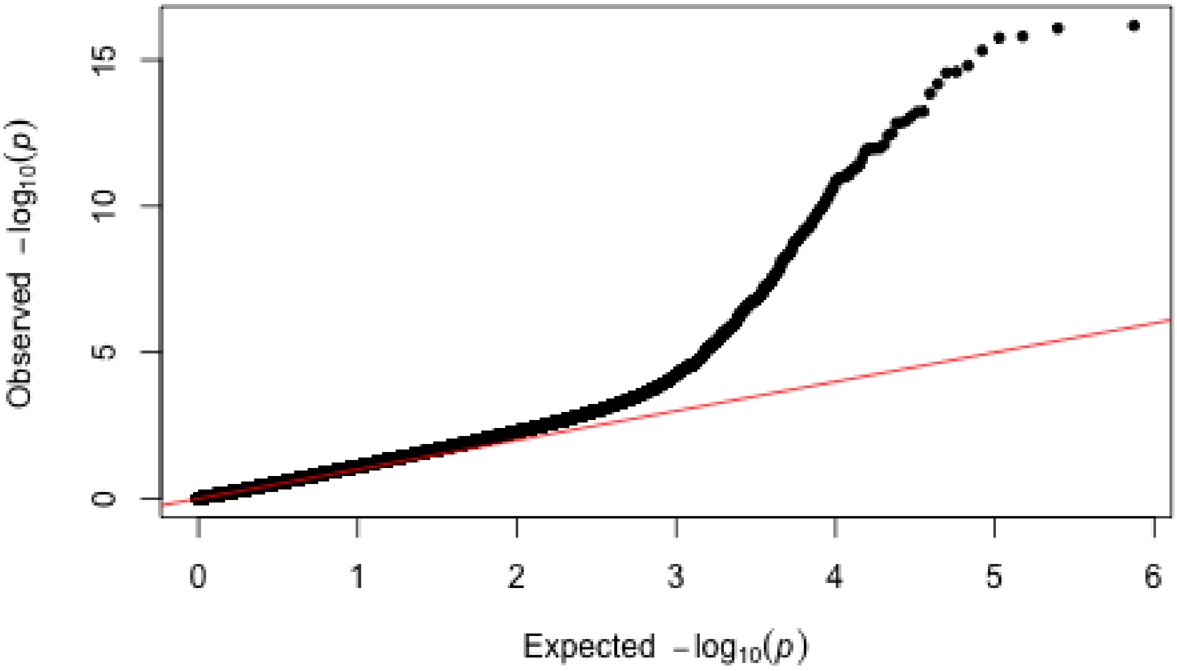
7)

Adjusting pvalue based on genome inflation factor

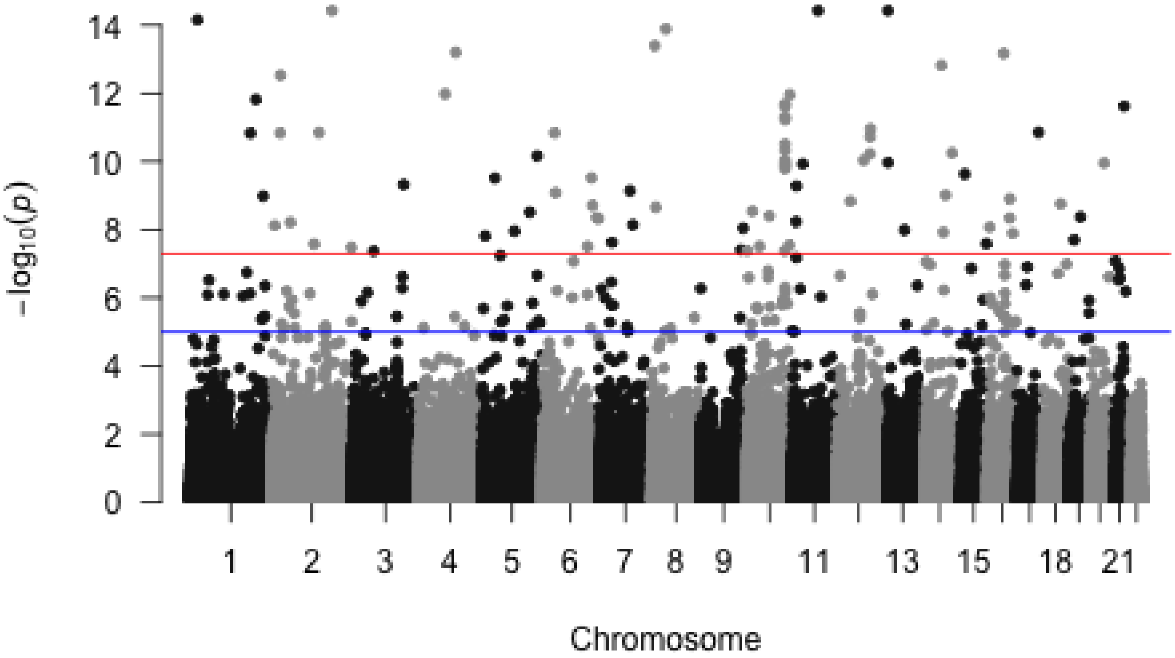
```
gwaResult <- gwaResult %>% mutate(PValue.adjust=pchisq(qchisq(1-PValue, 1)/genome.IF, 1, lower.tail = F)
```

QQPlot

```
qqman::qq(gwaResult$PValue)
```



```
manhattan.data <- gwaResult %>% select(SNP=rsid, CHR=Chrom, P=PValue.adjust, BP=pos)
manhattan(manhattan.data)
```



Genetic Architecture and Disease Similarity

1)

Read in the data file for all of the diseases

```
gwaResult.All <- read.delim("./results/gwa_results_all.csv", sep = "\t", header = T, stringsAsFactors =
gwaResult.All$Disease <-as.factor(gwaResult.All$Disease)
gwaResult.All$logPvalue <- -log10(gwaResult.All$PValue)
```

Filter out the data results

```
print(paste("SNPs before filter: ", nrow(gwaResult.All)))
```

```
## [1] "SNPs before filter:  3359657"
```

```
# Some SNPs have pvalues of 0
gwaResult.All <- gwaResult.All %>%  filter(PValue > 0 & HWPValue > 0)

# SNPs that deviate from Hardy-Weinberg because these are most likely due to genotyping errors
# and SNPs that have MAF < 0.01
gwaResult.All <- gwaResult.All %>%  filter(HWPValue >= 0.05 & ControlMinAlleleFreq >= 0.01 & DiseaseMin
print(paste("SNPs after filter: ", nrow(gwaResult.All)))
```

```
## [1] "SNPs after filter:  2631819"
```

Order the disease by number of significant associations found.

- PValues more than `1e-7` for each disease

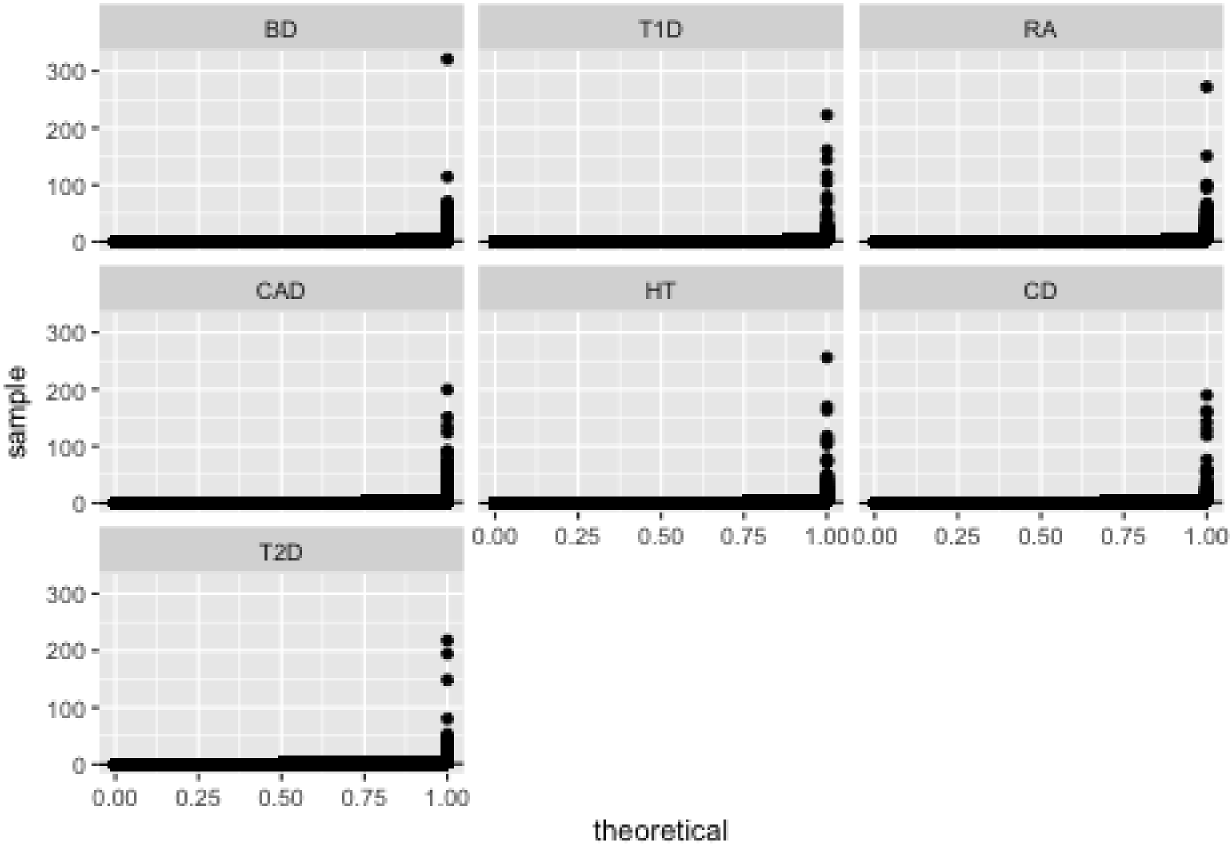
```
summ <- gwaResult.All %>% filter(PValue <= 1e-7) %>% group_by(Disease) %>% summarise(N=length(PValue),
summ
```

```
## # A tibble: 7 x 5
##   Disease     N                                              min     max      sd
##   <fct>   <int>                                            <dbl>   <dbl>   <dbl>
## 1 BD        231                                       Inf.NaNe-322 9.52e-8 2.04e-8
## 2 T1D       199                                          1.19e-223 9.30e-8 1.93e-8
## 3 RA        186                                          5.25e-273 9.04e-8 1.16e-8
## 4 CAD       180                                          3.03e-200 9.21e-8 1.55e-8
## 5 HT        177                                          7.47e-257 9.45e-8 1.52e-8
## 6 CD        164                                          2.80e-190 9.92e-8 2.41e-8
## 7 T2D       153                                          1.20e-218 9.98e-8 1.87e-8
```

```
gwaResult.All$Disease <- factor(gwaResult.All$Disease, levels=summ$Disease)
```

Do a GGPlot of the QQ plots for all diseases ordered by number of associations found.

```
ggplot(gwaResult.All) + geom_qq(mapping=aes(sample=logPvalue), distribution = qunif) + geom_abline(inte
```



2)

What diseases are associated with SNP rs6679677?

```
snp.rs6679677 <- gwaResult.All[which(gwaResult.All$rsid=='rs6679677'),]
snp.rs6679677 %>% select(Disease, PValue, rsid) %>% filter(PValue < 1e-7)
```

```
##   Disease       PValue      rsid
## 1      RA 5.559734e-28 rs6679677
## 2     T1D 5.506928e-27 rs6679677
```

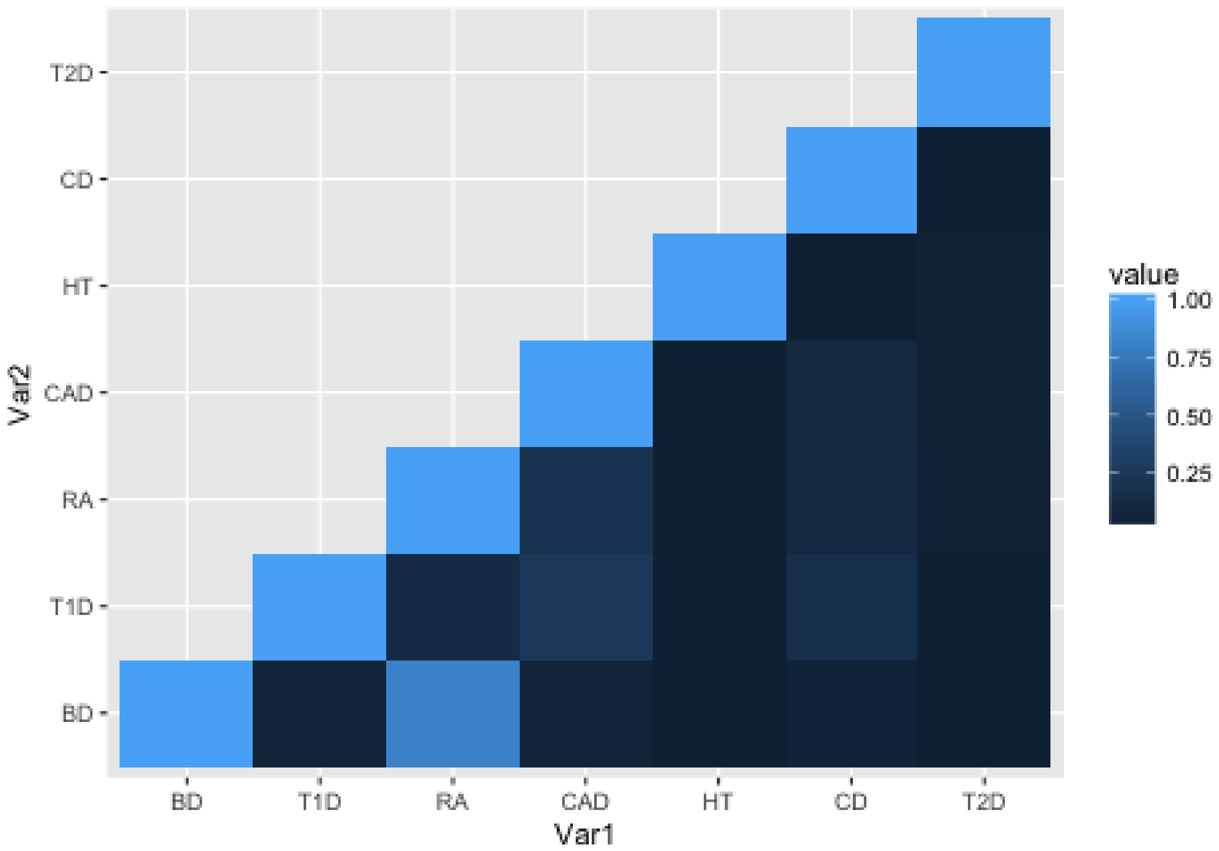
Type 1 Diabetes and Rheumatoid Arthritis are significantly associated with this SNP.

3)

If a SNP is found in multiple diseases, that means the gene that contains the SNP is involved in multiple biological pathways (ie. in a higher Gene Ontology), so it has effects towards a multitude of phenotypes. A good example of this is exhibited in autoimmune diseases that involve the MHC pathway towards broad autoimmune diseases.

4)

```r
corr.matrix <- gwaResult.All %>% select(Disease, OR) %>% split(.$Disease)

#resize to the smallest dataset so it can be converted to a data.frame
corr.matrix <- lapply(corr.matrix, function(x){
  x[1:375972,2]
})

corr.matrix <- as.data.frame(corr.matrix)
corr.values <- cor(corr.matrix)

lowerTri <- function(cormat){
    cormat[upper.tri(cormat)]<- NA
    return(cormat)
}

corr.values <- lowerTri(corr.values)
corr.values <- melt(corr.values, na.rm = T)

ggplot(data = corr.values, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()
```



From the plot, it looks like Bipolar Disorder and RA are most correlated in their OR for their SNPs with correlation value of 0.817477261.