GWAS Final Report

Jake Sauter

4/29/2021

Libraries Used

```
library(dplyr)
library(knitr)
library(ggplot2)
library(parallel)
library(magrittr)
library(patchwork)
```

Loading Data

Background

The data associated with the following analyses represents the mRNA expression levels of five different transcripts of interest (ERAP2, PEX6, FAHD1, GFM1 and MARCHF7), measured from four European or European-related populations. mRNA expression profiles were quantified through Next-Generation Sequencing (NGS) of reverse-transcribed cDNA of RNA samples dervied from Lymphoblastic Cell Lines (LCLs) generated from each individual in the study from the 1000 Genomes Project.

The phenoty for this study was generated by the Genetic European Variation in Health and Disease (gEUVADIS). A published Nature Article Transcriptome and genome sequencing uncovers functional variation in humans resulted from these data, and the data can be accessed via an International Genome Sample Resource Data Portal. The individuals included in the following analyses are apart of four seperate populations, **CEU** (Utah (U.S.A) residents with European ancestry), **FIN** (Finns), **GBR** (British) and **TSI** (Toscani).

The goal of the following analysis is to reproducilby locate positions of causal single nucleotide polymorphisms (SNPs) for the five LCL-mRNA related phenotypes aquired in the reference study. In order to accomplish this goal, a standard Genome Wide Association Study (GWAS) will be performed, in which independent hypothesis tests for association will be performed for each genotype-phenotype pair. Our aim is to identify genomic locations in which we calculate a significant association between a difference in genotype and observed phenotype, indicated by a significant ($\alpha < 0.05$) multiple testing corrected p-value for these locations. Due to the nature of the complex biological processes occuring with the information stored in genomic DNA, a significant p-value (for other than statitical reasons) may not indicate a true causual polymorphism is present at the exact location, more specifically due to linkage disequilibrium, or less formally observed correlation between closely related genomic positions.

To begin these analyses, an inspection of the reference study data will be performed, aiming to pinpoint missing, outlier, or addition trends in the data that break necessary assumptions in the analyses to follow. We will see that none of the prior mentioned abnormalities were present in the data. After data inpsection and validation, we inspect the distribution of phenotype data. The distributions of all measured RT-cDNA phenotypes appeared to be continuous and sufficiently normally distributed, allowing for analysis using linear regression. Lastly, we present the results of the five GWAS analyses by means of QQ plots (determining if the resultant p-value distribution is fairly uniform as expected) and both Global and Local Manhattan Plots. The localalized Manhattan plots are then overlayed to determine possible concordance or disconcordance

Model Setup

In order to analyze the given genetic data, we will make use of the qunantiative genetic model for a muliple regression.

Additive Genetic Coding

$$X_a(A_1A_1) = -1, X_a(A_1A_2) = 0, X_a(A_2A_2) = 1,$$

Dominant Genetic Coding

$$X_d(A_1A_1) = -1, X_d(A_1A_2) = 1, X_d(A_2A_2) = -1,$$

We will now use these encoded variables in our multiple regession model equation below

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

where ϵ $N(0, \sigma^2)$. This set of model and encodings allow us to moel the relationship between the genotype information (X_a, X_d) to the continous phenotype (Y), while preserving our genetic understanding of additive and dominant allele effects.

In the following analyses, we will use this model to perform a hypothesis test $H_0 : \beta_a = 0 \cap \beta_d = 0$ (indicating that additive or dominant genetic effects were not found to be assocaited with the phenotypic difference) vs $H_a : \beta_a \neq 0 \cup \beta_d \neq 0$ (indicating that either one or both of addiditive genetic effects are associated with the observed phenotypic difference).

Understanding the Data

Genentic Phenotype Information

Below we see the gene information of the 5 phenotypes encoded in our file

```
gene_info <-
  read.csv(file.path(data_path,
                     'gene_info.csv'))

gene_info %>%
  kable()
```

probe	chromosome	start	end	symbol
ENSG00000136536.9	2	159712456	159768582	MARCH7
ENSG00000180185.7	16	1827223	1840206	FAHD1
ENSG00000124587.9	6	42963872	42979242	PEX6
ENSG00000164308.12	5	96875939	96919702	ERAP2
ENSG00000168827.9	3	158644496	158692571	GFM1

Phenotypes

```
phenotypes <-
  read.csv(file.path(data_path, 'phenotypes.csv'),
                       row.names = 1, header = TRUE,
                       stringsAsFactors = FALSE) %>%
  set_colnames(c('ERAP2', 'PEX6', 'FAHD1', 'GFM1', 'MARCH7'))


phenotypes %>%
  dim()
```

```
[1] 344   5
```

```
phenotypes %>%
  names()
```

```
[1] "ERAP2"  "PEX6"   "FAHD1"  "GFM1"   "MARCH7"
```

```
phenotypes %>%
  head() %>%
  kable()
```

	ERAP2	PEX6	FAHD1	GFM1	MARCH7
HG00096	-1.166337	-0.6859358	-0.3062421	0.9275844	0.3600088
HG00097	-1.045803	-0.1789576	-1.0333305	-0.6676644	-1.7782530
HG00099	1.045803	-0.8945747	-2.5242603	-2.7590424	-2.5242603
HG00100	0.223446	-0.0982433	1.1521112	0.2533471	0.0545189
HG00101	-0.223446	0.8625121	1.3782830	1.0842344	0.2608569
HG00102	-1.457685	-0.3291676	-0.1863454	-1.1663369	-2.3783289

What Genotypes are Present?

Below we see that we have **50,000** genotype markers for each of our **344** individuals. Each individual is represented in a **row** in our `genotype` data frame, with each **column** corresponding the the number of minor alleles the individual has present in their DNA.

```
genotypes <-
  read.csv(file.path(data_path, 'genotypes.csv'),
           row.names = 1, header = TRUE,
           stringsAsFactors = FALSE)
```

```
genotypes %>%
  dim()
```

```
[1]   344 50000
```

```
genotypes %>%
  .[1:10, 1:5] %>%
  kable()
```

	rs10399749	rs62641299	rs115523412	rs75932129	rs10900604
HG00096	0	2	0	1	1
HG00097	0	2	0	2	0
HG00099	1	1	0	0	1
HG00100	0	2	0	2	0
HG00101	1	2	0	1	1
HG00102	0	2	0	1	0
HG00103	0	2	0	2	0
HG00104	0	2	0	1	0
HG00106	1	2	0	1	0
HG00108	1	2	0	2	0

Additive and Dominant Genetic Encoding of Genotype Information

We can then convert this encoding to our standard X_a and X_d encodings (additive and dominant genetic codings) in the following way.

```
Xa <- as.data.frame(genotypes) - 1
Xd <- 1 - 2*abs(Xa)
```

SNP Info

```
snp_info <-
  read.csv(file.path(data_path,
                     'SNP_info.csv'))

snp_info %>%
  head() %>%
  kable()
```

chromosome	position	id
1	55298	rs10399749
1	79049	rs62641299
1	826577	rs115523412
1	861386	rs75932129
1	863019	rs10900604
1	875399	rs58686784

Which chromosomes are represented in our samples?

```
snp_info %>%
  .$chromosome %>%
  unique()
```

```
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
```

Covariate Information

Below, we see that 2 covariates were measured in this study, **Population** and **Sex**.

```
covars <-
  read.csv(file.path(data_path,
                     'covars.csv'),
           row.names = 1)

covars %>%
  head() %>%
  kable()
```

	Population	Sex
HG00096	GBR	MALE
HG00097	GBR	FEMALE
HG00099	GBR	FEMALE
HG00100	GBR	FEMALE
HG00101	GBR	MALE
HG00102	GBR	FEMALE

Below we see that each covariate controlled population has over **30** samples, indicating that we should have enough data/information here in order to control for both covariates with introducing non-robust model characteristics.

```
covars %>%
  count(Population, Sex) %>%
  kable()
```

Population	Sex	n
CEU	FEMALE	37
CEU	MALE	41
FIN	FEMALE	55
FIN	MALE	34
GBR	FEMALE	46
GBR	MALE	39
TSI	FEMALE	43
TSI	MALE	49

Checking the Phenotype Data

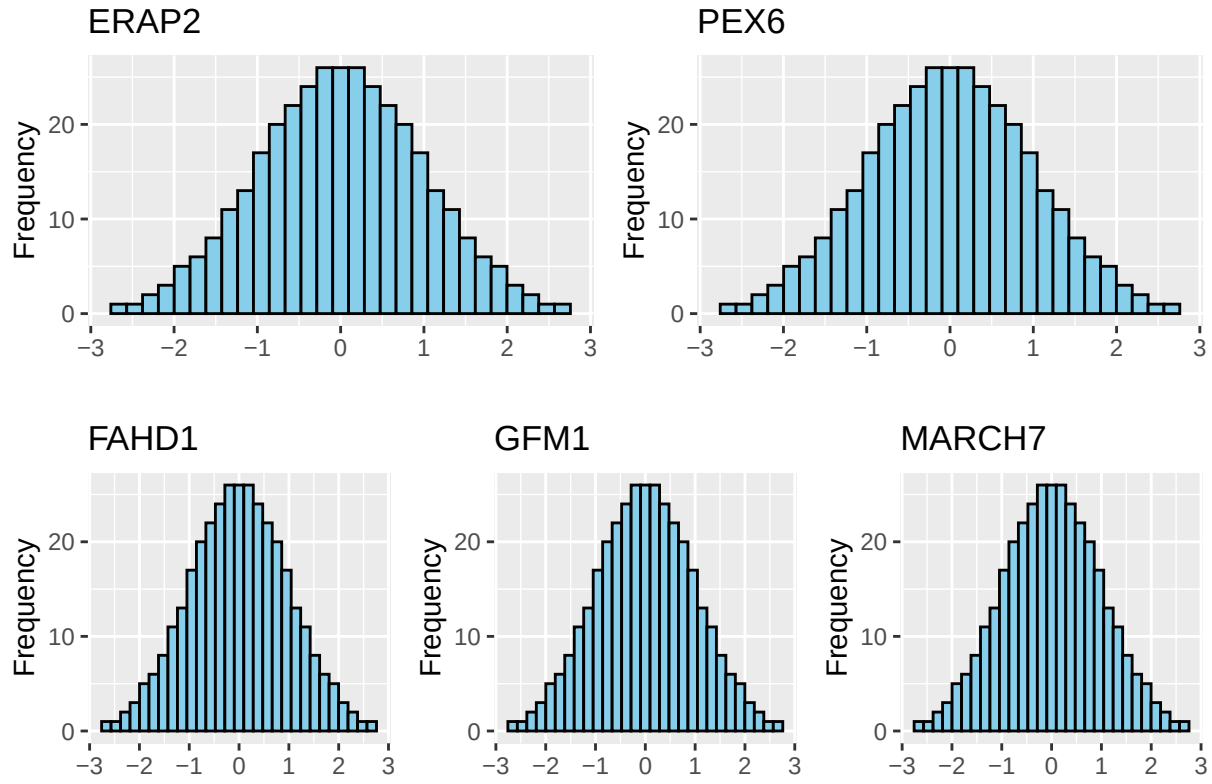
From the information above, it is clear that we are dealing with **5** phenotypes from **344** individuals, each phenotype describing the **continuous** variable of mRNA expression of the genes of interest.

Phenotype Histograms

All phenotypes appear to be be normally distributed between -3 and 3. This fits our expection of Normality in GWAS

TODO: Why does the phenotype need to be normal in a GWAS?

```
plot_list <- vector('list', 5)
names(plot_list) <- names(phenotypes)

for (phenotype in names(phenotypes)) {
  plot_list[[phenotype]] <-
    phenotypes %>%
      ggplot() +
      geom_histogram(aes_string(x = phenotype),
                     color = 'black',
                     fill = 'skyblue') +
    xlab('') + ylab('Frequency') +
    ggtitle(phenotype)
}

(plot_list[[1]] + plot_list[[2]]) /
  (plot_list[[3]] + plot_list[[4]] + plot_list[[5]])
```



Checking for Odd Phenotype Values

There does not seem to be any `NA` or outlier phenotypic values present in the data.

```
phenotypes %>%
  anyNA()
```

```
[1] FALSE
```

```
phenotypes %>%
  summary()
```

```
     ERAP2              PEX6              FAHD1              GFM1        
 Min.   :-2.7590   Min.   :-2.7590   Min.   :-2.7590   Min.   :-2.7590  
 1st Qu.:-0.6699   1st Qu.:-0.6699   1st Qu.:-0.6699   1st Qu.:-0.6699  
 Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000  
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  
 3rd Qu.: 0.6699   3rd Qu.: 0.6699   3rd Qu.: 0.6699   3rd Qu.: 0.6699  
 Max.   : 2.7590   Max.   : 2.7590   Max.   : 2.7590   Max.   : 2.7590  
     MARCH7        
 Min.   :-2.7590  
 1st Qu.:-0.6699  
 Median : 0.0000  
 Mean   : 0.0000  
```

```
 3rd Qu.: 0.6699
 Max.   : 2.7590
```

Check and Filter Genotype Data

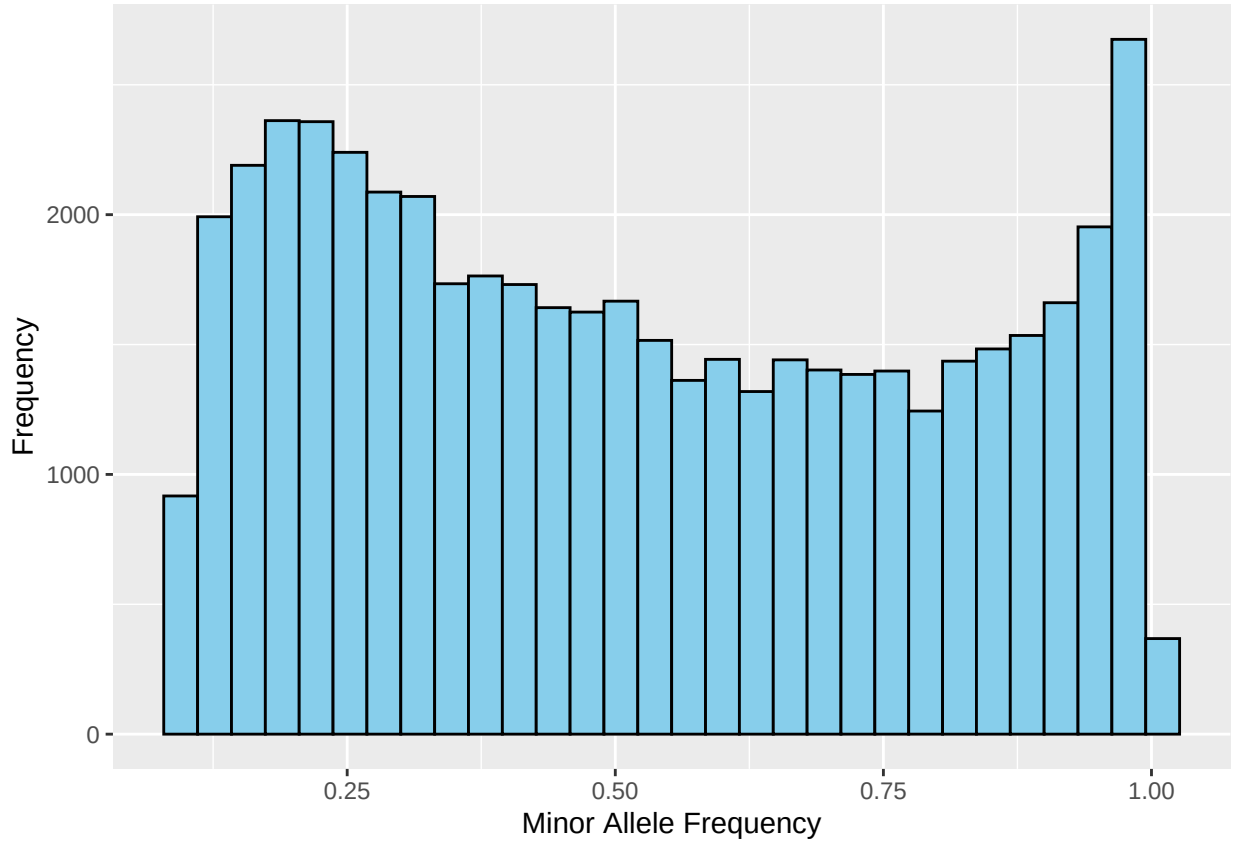
Data Quality Checks

Filter genotypes:

- Remove **individuals** with more than 10% missing data across all genotypes
- Remove **genotypes** with more than 5% missing data across all individuals.
- Remove **genotypes** with an MAF less than 5%

From the information below, we can see that no genotypes with an MAF of less than 5% are present in our data.

```
MAFs <-
  genotypes %>%
  apply(2, function(x) {
    length(x[x > 0]) / length(x)
  })

MAFs %>%
  data.frame(MAF = .) %>%
  ggplot() +
    geom_histogram(aes(x = MAF),
                   color = 'black',
                   fill = 'skyblue') +
    xlab('Minor Allele Frequency') +
    ylab('Frequency')
```




```
MAFs %>%
  .[. < 0.05] %>%
  length()
```

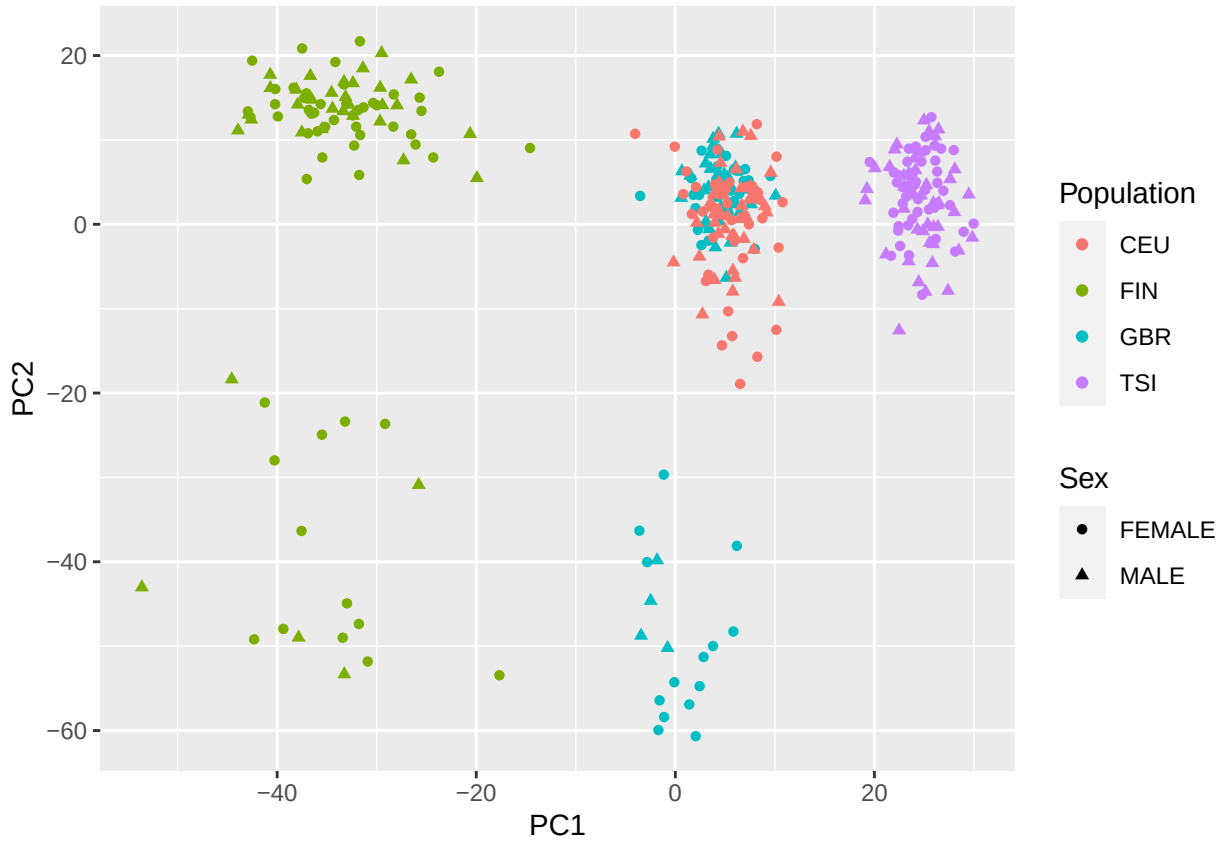
```
[1] 0
```

TODO: Hardy-Weinburg Equilibrium?

Principle Components Analysis

```
pca_result <-
  prcomp(genotypes,
         scale = TRUE,
         center = TRUE)
```

```
data.frame(PC1 = pca_result$x[, 1],
           PC2 = pca_result$x[, 2],
           Population = covars$Population,
           Sex = covars$Sex) %>%
ggplot() +
  geom_point(aes(x = PC1, y = PC2,
                 color = Population,
                 shape = Sex))
```



Performing GWAS Without Covariates

P-value Calculations

```
perform_gwas_no_covs <- function(genotypes, phenotype) {
  n_genotypes <- ncol(genotypes)

  gwas_results <-
    mclapply(seq_len(n_genotypes),
      function(genotype) {
        model <- lm(phenotype ~ Xa[,genotype] +
                                Xd[,genotype])

        p_val <-
          summary(model) %>%
          .$fstatistic %>%
          {pf(.[1], .[2], .[3],
              lower.tail = FALSE)}

        names(p_val) <-
          colnames(genotypes)[genotype]

        p_val
```

```
      }, mc.cores = 7)

  unlist(gwas_results)
}
```

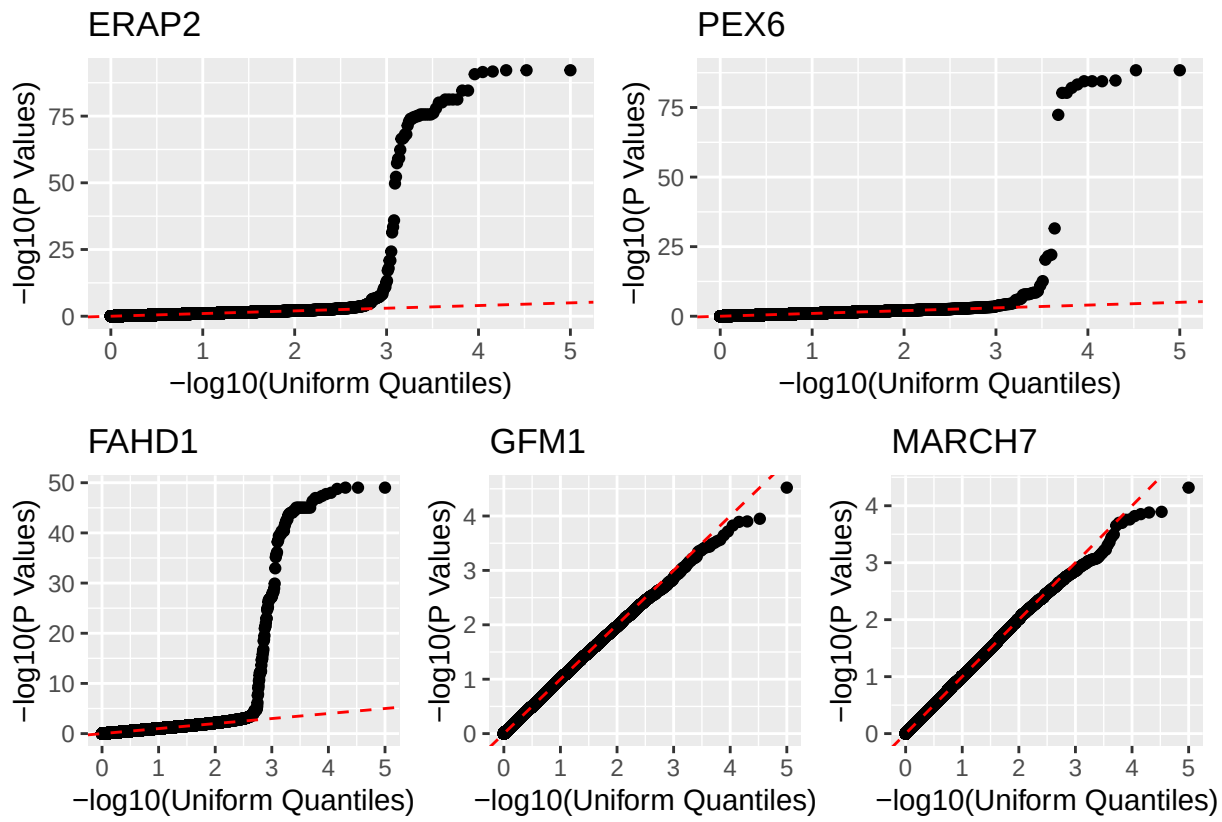
```
gwas_no_covs_results <-
  lapply(seq_len(ncol(phenotypes)),
         function(i) {
           perform_gwas_no_covs(genotypes,
                                phenotypes[,i])
   }) %>%
  set_names(colnames(phenotypes))
```

QQ Plots

```
plot_uniform_qq <- function(p_values, pheno_name) {

  uniform_quantiles <-
    qunif(ppoints(length(p_values)))

  p <-
    data.frame(p_values          = sort(p_values),
               uniform_quantiles = sort(uniform_quantiles)) %>%
    ggplot() +
      geom_point(aes(x = -log10(uniform_quantiles),
                     y = -log10(p_values))) +
      geom_abline(intercept = 0,
                  slope = 1,
                  color="red",
                  lty = 2) +
      xlab('-log10(Uniform Quantiles)') +
      ylab('-log10(P Values)') +
      ggtitle(pheno_name)

  p
}
```

```
phenotype_names <- names(gwas_no_covs_results)

plot_list <-
  lapply(seq_along(gwas_no_covs_results),
         function(i) {
           plot_uniform_qq(gwas_no_covs_results[[i]],
                           phenotype_names[[i]])
         })

(plot_list[[1]] + plot_list[[2]]) /
```

```
  (plot_list[[3]] + plot_list[[4]] + plot_list[[5]])
```



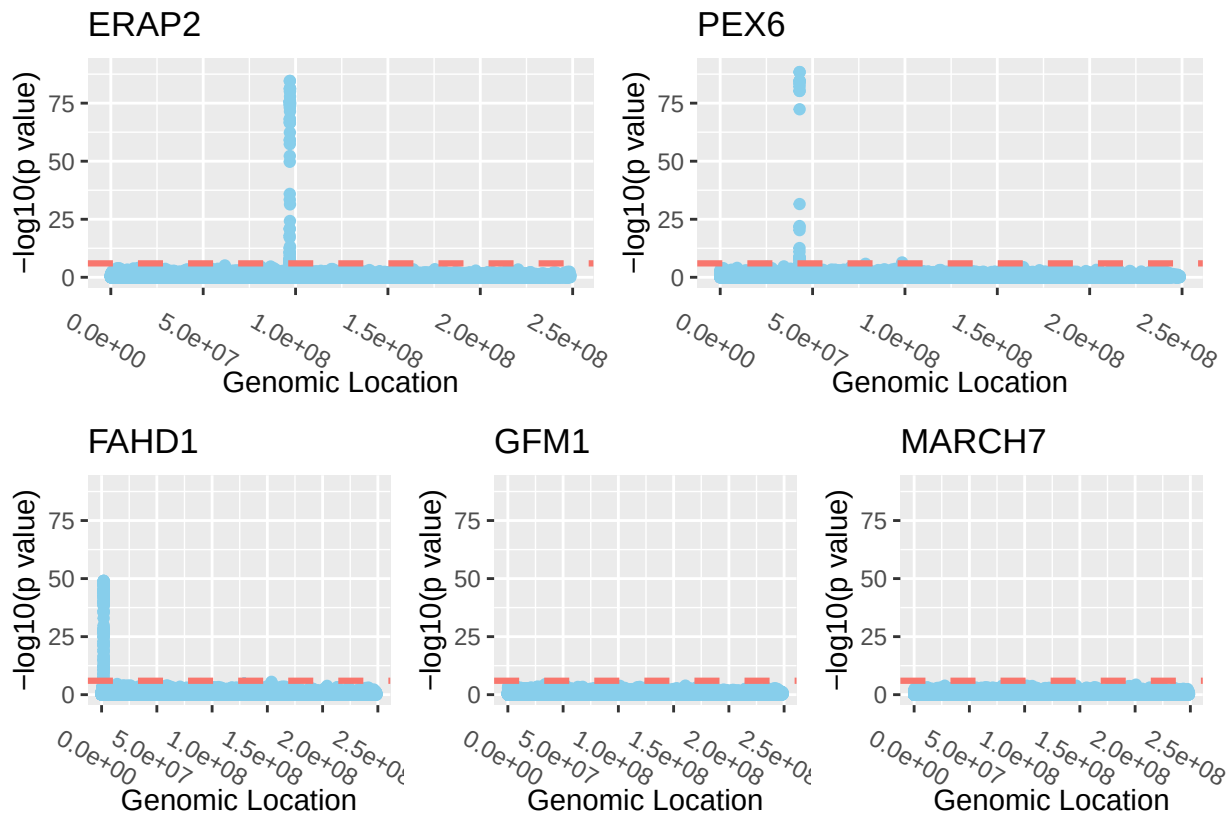
Manhattan Plots

```
## TODO modify to use the gene info to plot the location
# of the SNP in the genome, not the number the p-value is

plot_manhattan <- function(p_values, plot_title,
                           start = NULL, end = NULL) {


  ## Correlate p-values to genomic locations with snp_info
  unique_chromosomes <-
    snp_info$chromosome %>%
    unique()

  chromosome_offsets <-
    sapply(unique_chromosomes,
      function(chromosome) {
        offset <-
          max(snp_info[snp_info['chromosome'] == chromosome,
                       'position'])
```

```
        names(offset) <- chromosome
        offset
    })

  genomic_locations <-
    snp_info[snp_info['id'] == names(p_values), ]  %>%
    mutate(position =
             if_else(chromosome == 1,
                     position,
                     position + sum(chromosome_offsets[1:chromosome-1]))) %>%
    .$position

  n_genotypes <- length(p_values)

  df <-
    data.frame(
      genome_location = genomic_locations,
      pval = p_values) %>%
    mutate(plot_pval = -log10(pval))

  # localizing manhattan plot to better show
  # hits in the provided range
  if(!is.null(start) && !is.null(end)) {
    df <- df %>%
      filter(genome_location >= start &
             genome_location <= end)
  }


  p <-
    df %>%
    ggplot() +
    geom_point(aes(x = genome_location, y = plot_pval),
               col = 'skyblue') +
    geom_hline(aes(yintercept = -log10(0.05 / n_genotypes),
               color = 'red'), lty = 2, lwd = 1.1) +
    xlab('Genomic Location') + ylab('-log10(p value)') +
    ggtitle(plot_title) + ylim(c(0, 90)) +
    scale_x_continuous(labels = function(x) format(x, scientific = TRUE)) +
    theme(legend.position = 'none',
          axis.text.x = element_text(angle = -30))

  p
}
```

```
plot_list <-
  lapply(seq_along(gwas_no_covs_results),
```

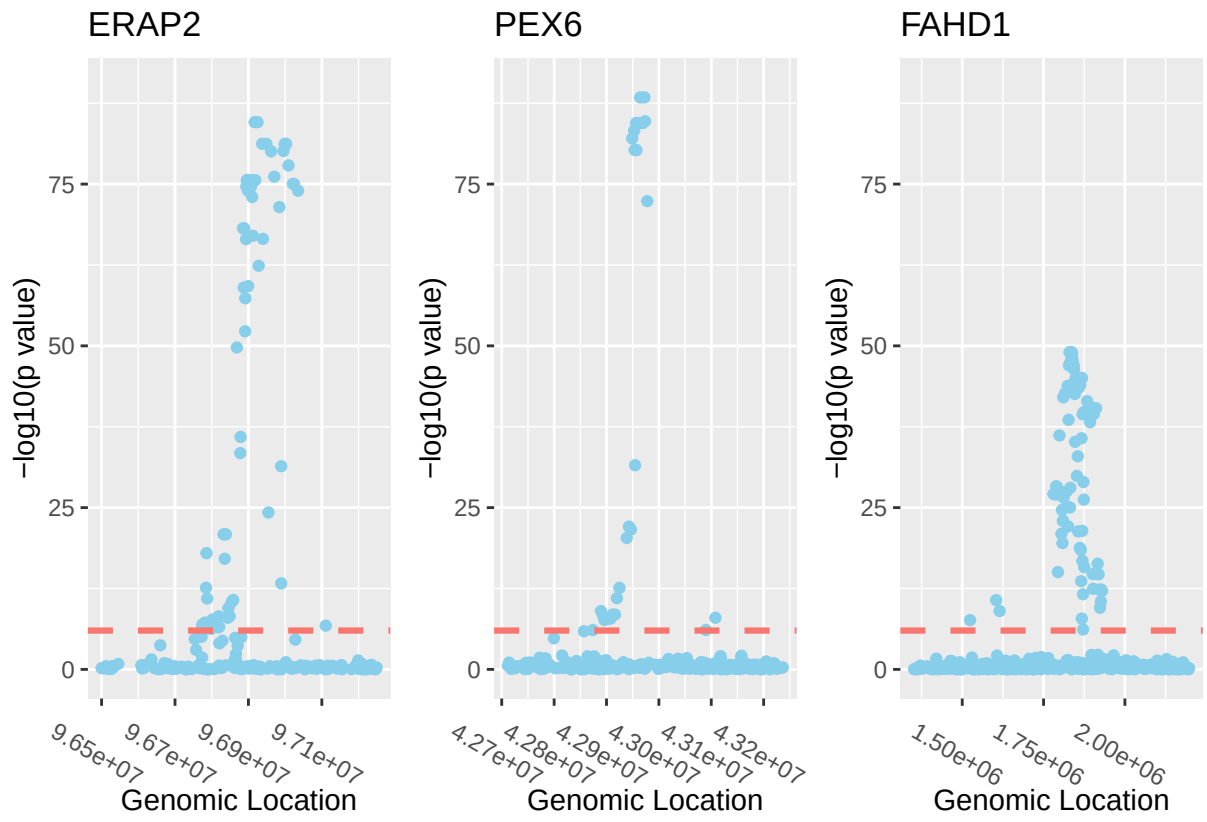
```
      function(i) {
        gwas_result <- gwas_no_covs_results[[i]]
        plot_name <- names(gwas_no_covs_results)[[i]]
        plot_manhattan(gwas_result, plot_name)
    })

(plot_list[[1]] + plot_list[[2]]) /
  (plot_list[[3]] + plot_list[[4]] + plot_list[[5]])
```



Of the three phenotypes that we have observed significant hits for, we will take a closer look at the local regions where the hits were found to be.

```
plot_manhattan(gwas_no_covs_results[['ERAP2']],
               'ERAP2',
               start = 9.65e7,
               end = 9.725e7) +
plot_manhattan(gwas_no_covs_results[['PEX6']],
               'PEX6',
               start = 4.27e7,
               end = 4.325e7) +
plot_manhattan(gwas_no_covs_results[['FAHD1']],
               'FAHD1',
               start = 1.35e6,
```

```
                end = 2.2e6)
```



Performing GWAS With Covariates

P-value Calculations

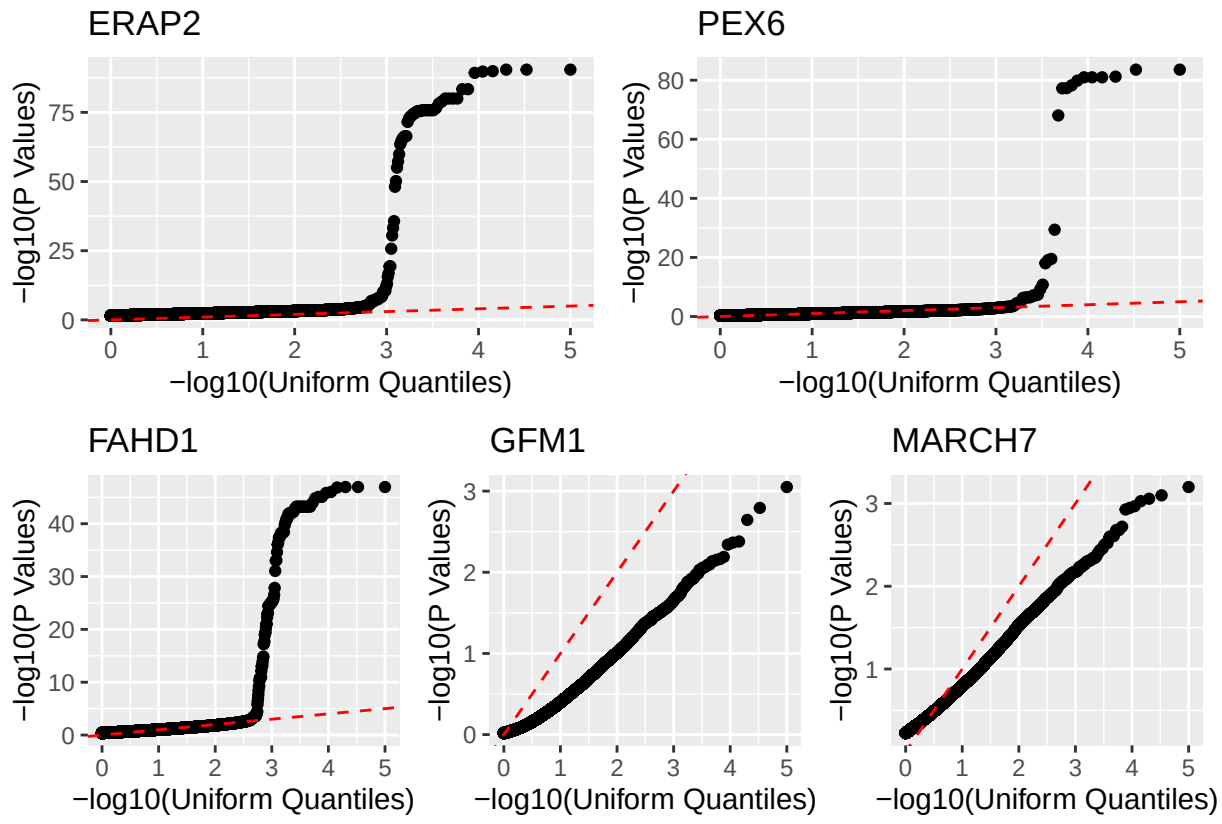
```
perform_gwas_with_covs <- function(genotypes, phenotype) {
  n_genotypes <- ncol(genotypes)

  gwas_results <-
    mclapply(seq_len(n_genotypes),
      function(genotype) {
        model <- lm(phenotype ~ Xa[,genotype] +
                                Xd[,genotype] +
                                covars$Sex +
                                covars$Population)

        p_val <-
          summary(model) %>%
          .$fstatistic %>%
          {pf(.[1], .[2], .[3],
              lower.tail = FALSE)}
```

```
        names(p_val) <-
          colnames(genotypes)[genotype]

        p_val
      }, mc.cores = 7)

  unlist(gwas_results)
}
```

```
gwas_with_covs_results <-
  lapply(seq_len(ncol(phenotypes)),
         function(i) perform_gwas_with_covs(genotypes, phenotypes[,i])) %>%
  set_names(colnames(phenotypes))
```

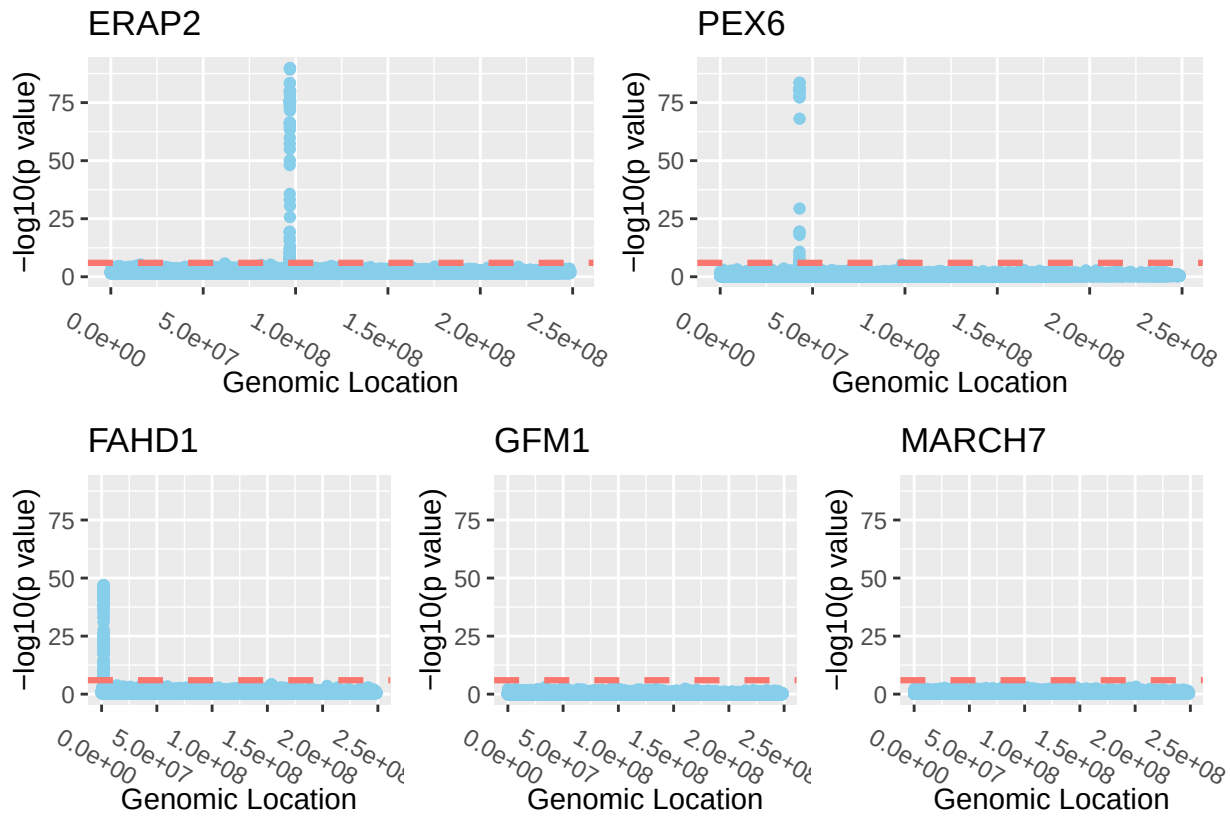
QQ Plots

```
phenotype_names <- names(gwas_with_covs_results)

plot_list <-
  lapply(seq_along(gwas_with_covs_results),
         function(i) {
           plot_uniform_qq(gwas_with_covs_results[[i]],
                           phenotype_names[[i]])
         })

(plot_list[[1]] + plot_list[[2]]) /
  (plot_list[[3]] + plot_list[[4]] + plot_list[[5]])
```



Manhattain Plots

```
plot_list <-
  lapply(seq_along(gwas_with_covs_results),
      function(i) {
        gwas_result <- gwas_with_covs_results[[i]]
        plot_name <- names(gwas_with_covs_results)[[i]]
        plot_manhattan(gwas_result, plot_name)
    })

(plot_list[[1]] + plot_list[[2]]) /
  (plot_list[[3]] + plot_list[[4]] + plot_list[[5]])
```



Of the three phenotypes that we have observed significant hits for, we will take a closer look at the local regions where the hits were found to be. We see below that we have found the same hits with and without covariates included in our model.

```
plot_manhattan(gwas_with_covs_results[['ERAP2']],
               'ERAP2',
               start = 9.65e7,
               end = 9.725e7) +
plot_manhattan(gwas_with_covs_results[['PEX6']],
               'PEX6',
               start = 4.27e7,
               end = 4.325e7) +
plot_manhattan(gwas_with_covs_results[['FAHD1']],
               'FAHD1',
               start = 1.35e6,
               end = 2.2e6)
```

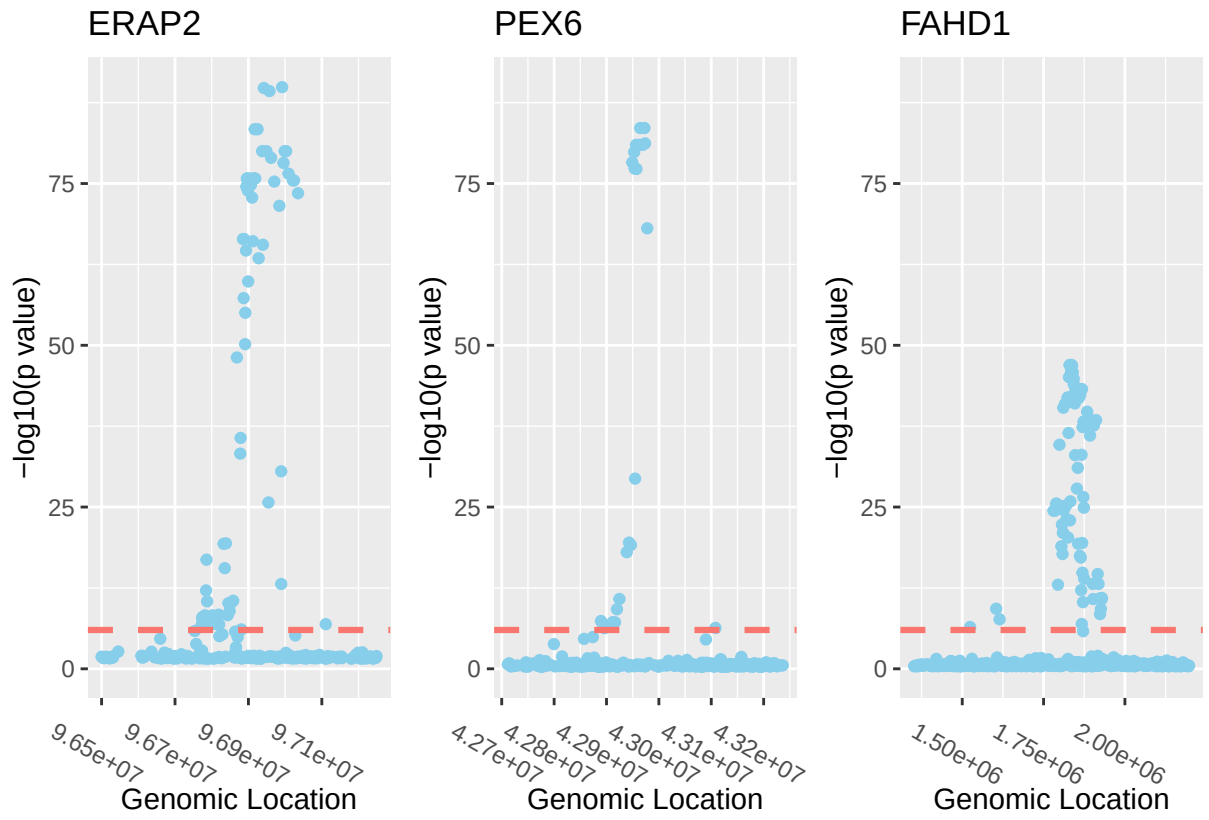


Table of Most Significant Hits

```
data.frame(
  Phenotype = c('ERAP2', 'PEX6', 'FAHD1'),
  '# Sig. Genotypes' = c(1, 2, 3),
  'Top 3 Sig. Genotypes' = c('a,b,c',
                             'a,b,c',
                             'a,b,c')
  )%>%
  kable()
```

Phenotype	X..Sig..Genotypes	Top.3.Sig..Genotypes
ERAP2	1	a,b,c
PEX6	2	a,b,c
FAHD1	3	a,b,c

Biological Support of Found Hits

Literature Review / GeneCards