GWAS of Threespine Stickleback Kinematics

Sara J Smith

09 May, 2025

```
library(qqman)
library(kableExtra)
library(reshape2)
library(viridis)
library(tidyverse)
```

Read in data from each of the GWAS analyses, clean, and save as a new object.

Continuous GWAS analyses:

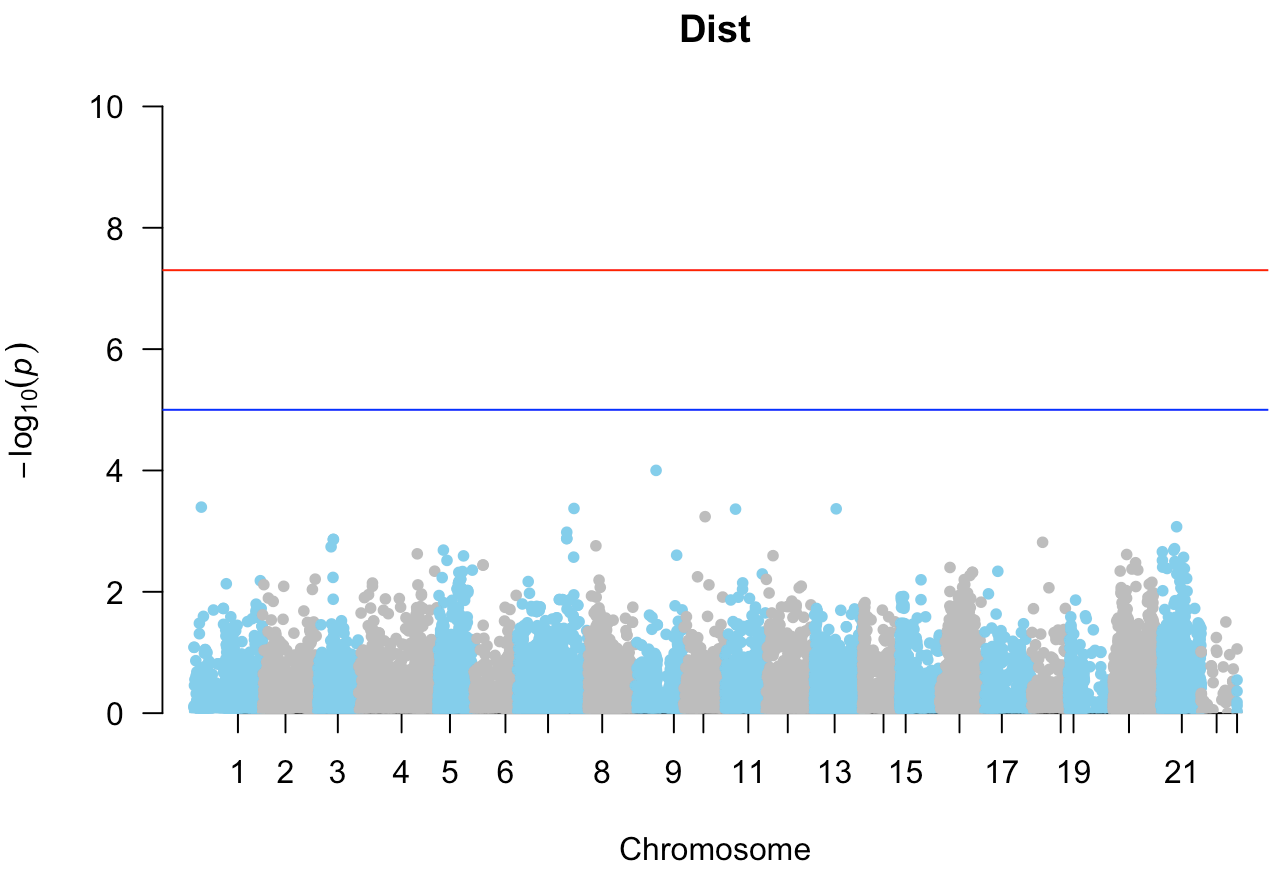
```
file_dir <- "~/Desktop/MRU_Faculty/Research/ucr_stickles/gwas_results"
file_paths <- list.files(file_dir, pattern = "gasAcu.plink.*.glm.linear", full.names = T
RUE)


for (file in file_paths) {

    name <- str_split(basename(file), "\\.", simplify = TRUE)[, 3]

    data <- read.table(file, col.names = c("CHR", "POS", "ID", "REF", "ALT", "A1",
        "TEST", "OBS_CT", "OR", "LOG(OR)_SE", "Z_STAT", "P")) %>%
        dplyr::select(CHR, POS, ID, P) %>%
        filter((CHR <= 22 | is.na(CHR))) %>%
        filter(CHR != 1.1 | is.na(CHR)) %>%
        filter(CHR != 21.1 | is.na(CHR)) %>%
        mutate(CHR = replace_na(CHR, 23)) %>%
        rename(chr = CHR, pos = POS, p = P)

    assign(name, data, envir = .GlobalEnv)
}
```

Binary GWAS analysis:

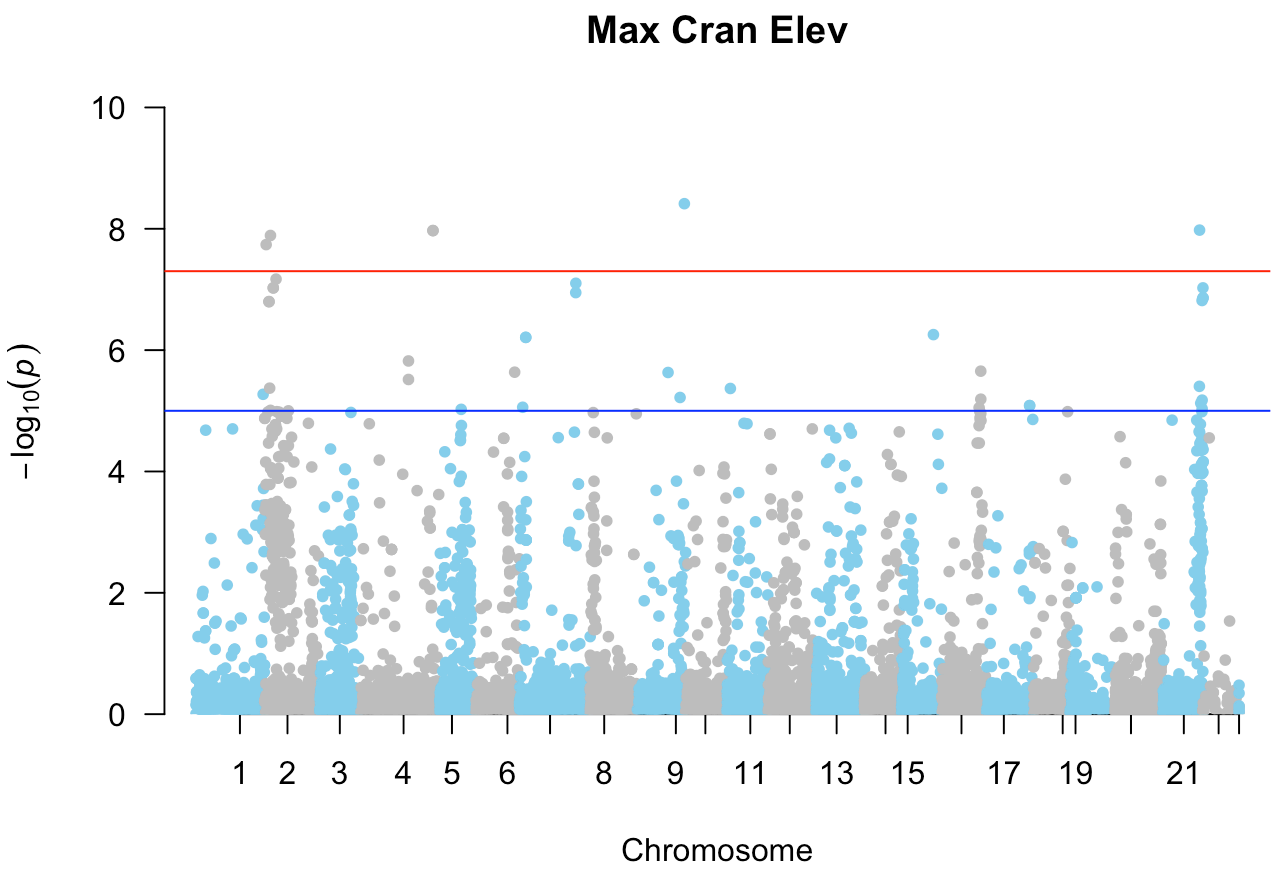
```
sf <- read.table("gwas_results/gasAcu.plink.sf.glm.logistic", col.names = c("CHR",
    "POS", "ID", "REF", "ALT", "A1", "TEST", "OBS_CT", "OR", "LOG(OR)_SE", "Z_STAT",
    "P")) %>%
    dplyr::select(CHR, POS, ID, P) %>%
    filter((CHR <= 22 | is.na(CHR))) %>%
    filter(CHR != 1.1 | is.na(CHR)) %>%
    filter(CHR != 21.1 | is.na(CHR)) %>%
    mutate(CHR = replace_na(CHR, 23)) %>%
    rename(chr = CHR, pos = POS, p = P)
```

Make Manhattan plots for each phenotype (with a red genome-wide significance line of 5e-8 and a blue suggestive significance line of 1e-5) & pull SNPs with genome-wide significance

```
threshold <- 5e-08

manhattan(dist, main = "Dist", chr = "chr", bp = "pos", p = "p", snp = "ID", ylim = c(0,
    10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # no genome-wide sig
(GWS)
```



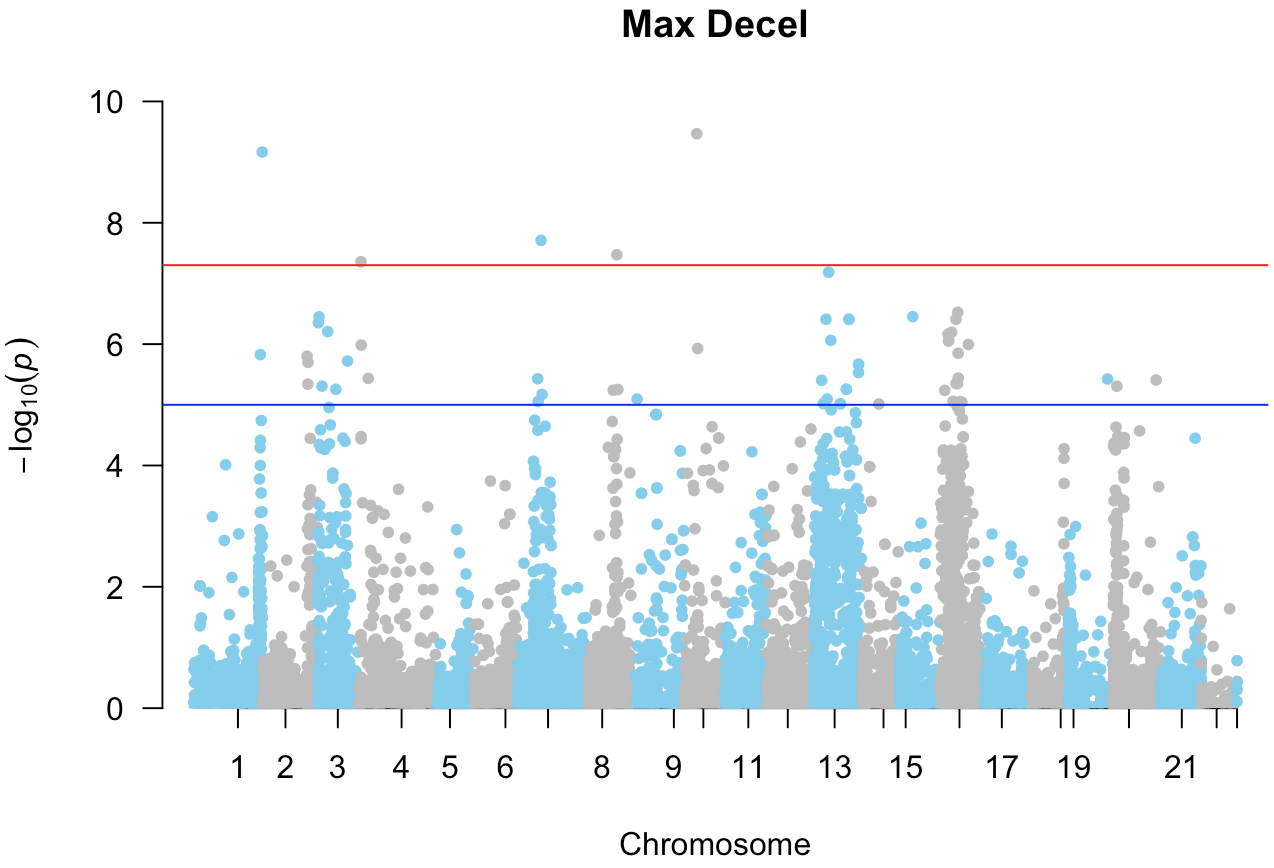

```
manhattan(maxCranElev, main = "Max Cran Elev", chr = "chr", bp = "pos", p = "p",
    snp = "ID", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))  # genome wide sig
```




```
mce_sig <- maxCranElev %>%
    filter(p < threshold)

maxCranElev %>%
    filter(p < threshold) %>%
    kbl() %>%
    kable_minimal()
```

chr	pos	ID	p
2	622333	2:622333	0
2	2475047	2:2475047	0
4	6436533	4:6436533	0
4	31001914	4:31001914	0
4	31001974	4:31001974	0
9	17892251	9:17892251	0
9	20267308	9:20267308	0
16	16332828	16:16332828	0
16	16417317	16:16417317	0
16	16674620	16:16674620	0
16	16732849	16:16732849	0
17	19690144	17:19690144	0
21	15879518	21:15879518	0
21	16656917	21:16656917	0

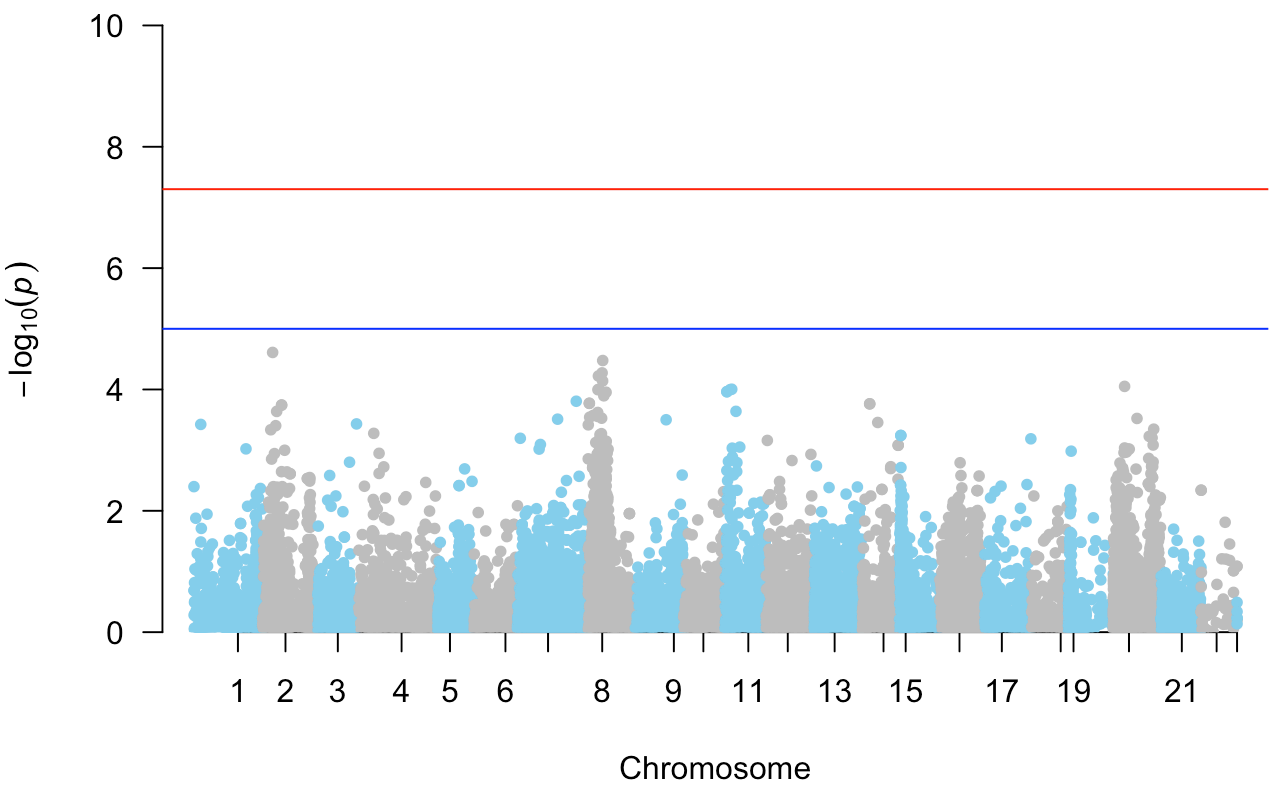
```
manhattan(maxDecel, main = "Max Decel", chr = "chr", bp = "pos", p = "p", snp = "ID",
    ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # GWS
```



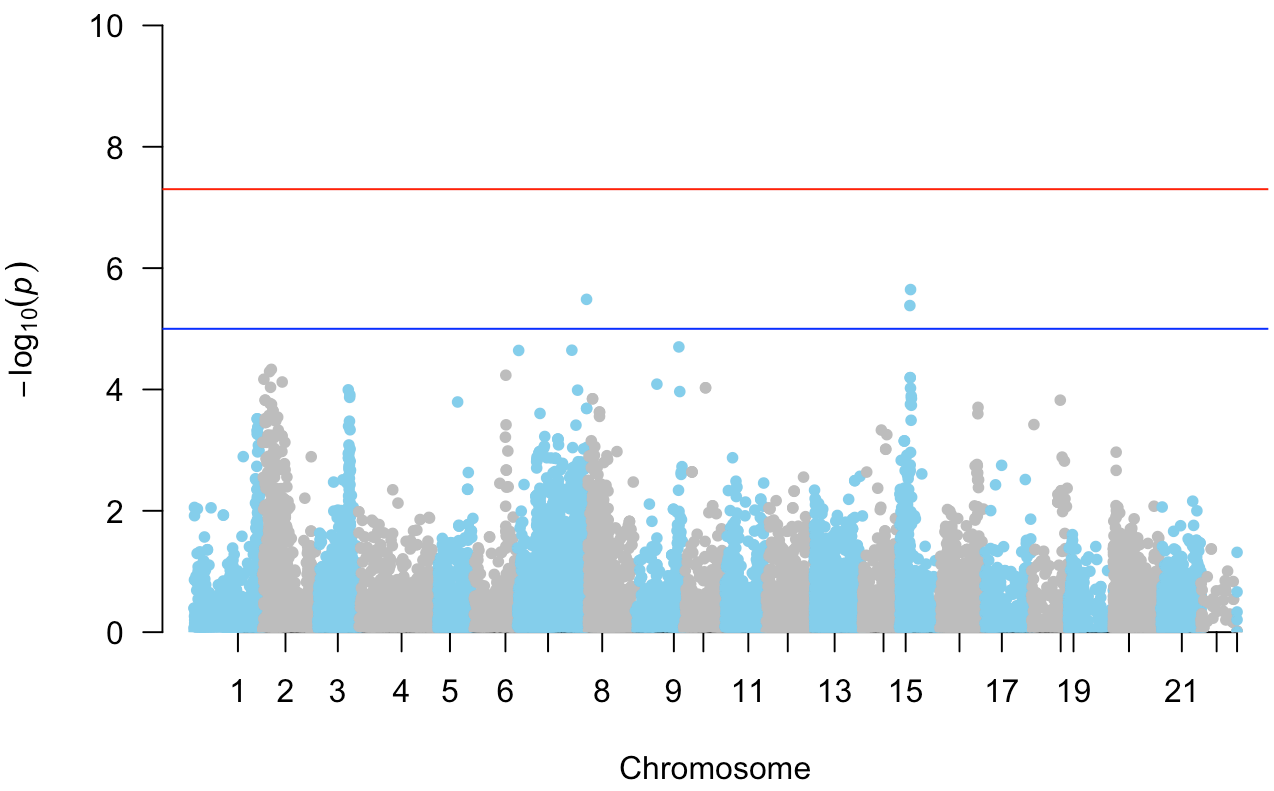

```
md_sig <- maxDecel %>%
    filter(p < threshold)

maxDecel %>%
    filter(p < threshold) %>%
    kbl() %>%
    kable_minimal()
```

chr	pos	ID	p
1	29397586	1:29397586	0
4	960973	4:960973	0
7	10254425	7:10254425	0
8	10827157	8:10827157	0
8	12422393	8:12422393	0
10	5678087	10:5678087	0

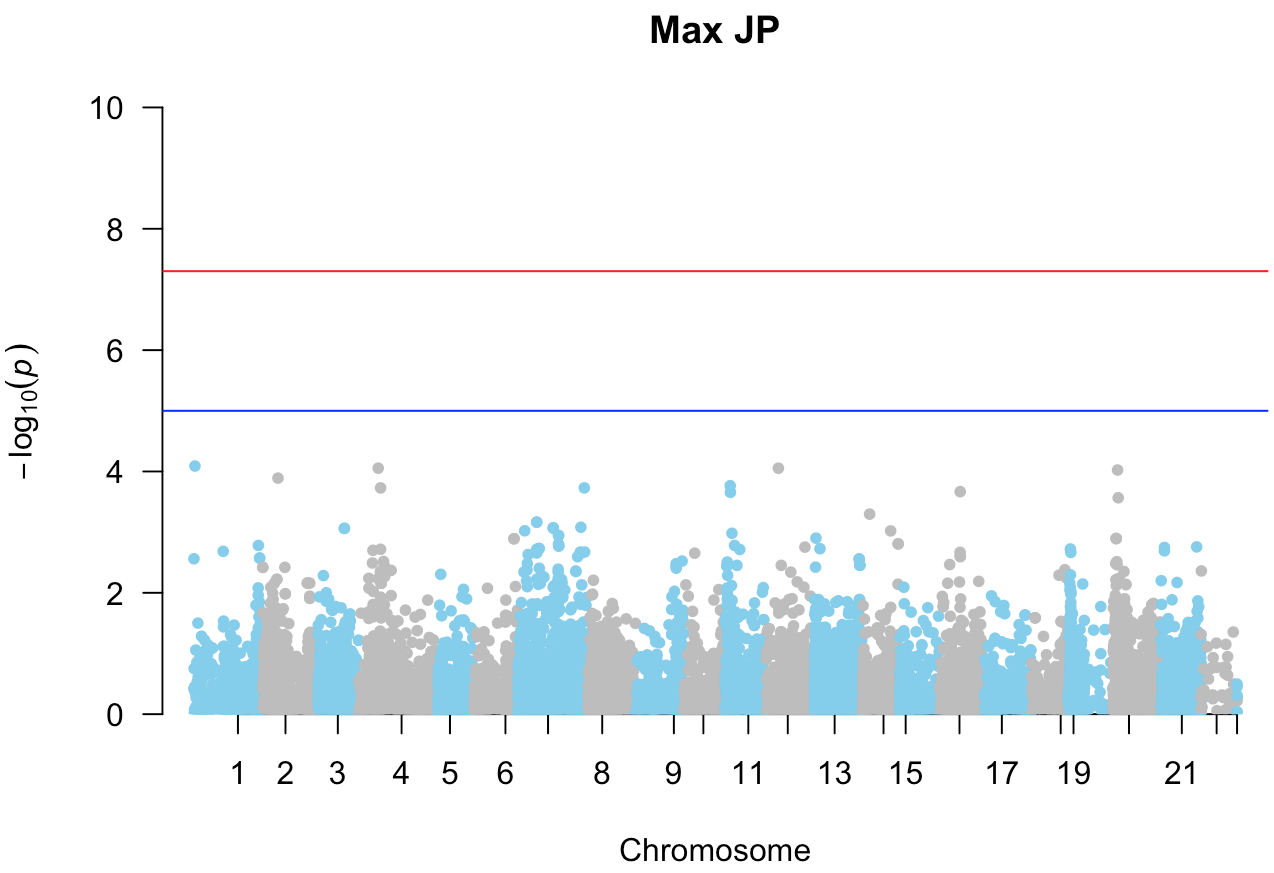
```
manhattan(maxGape, main = "Max Gape", chr = "chr", bp = "pos", p = "p", snp = "ID",
    ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # no GWS
```



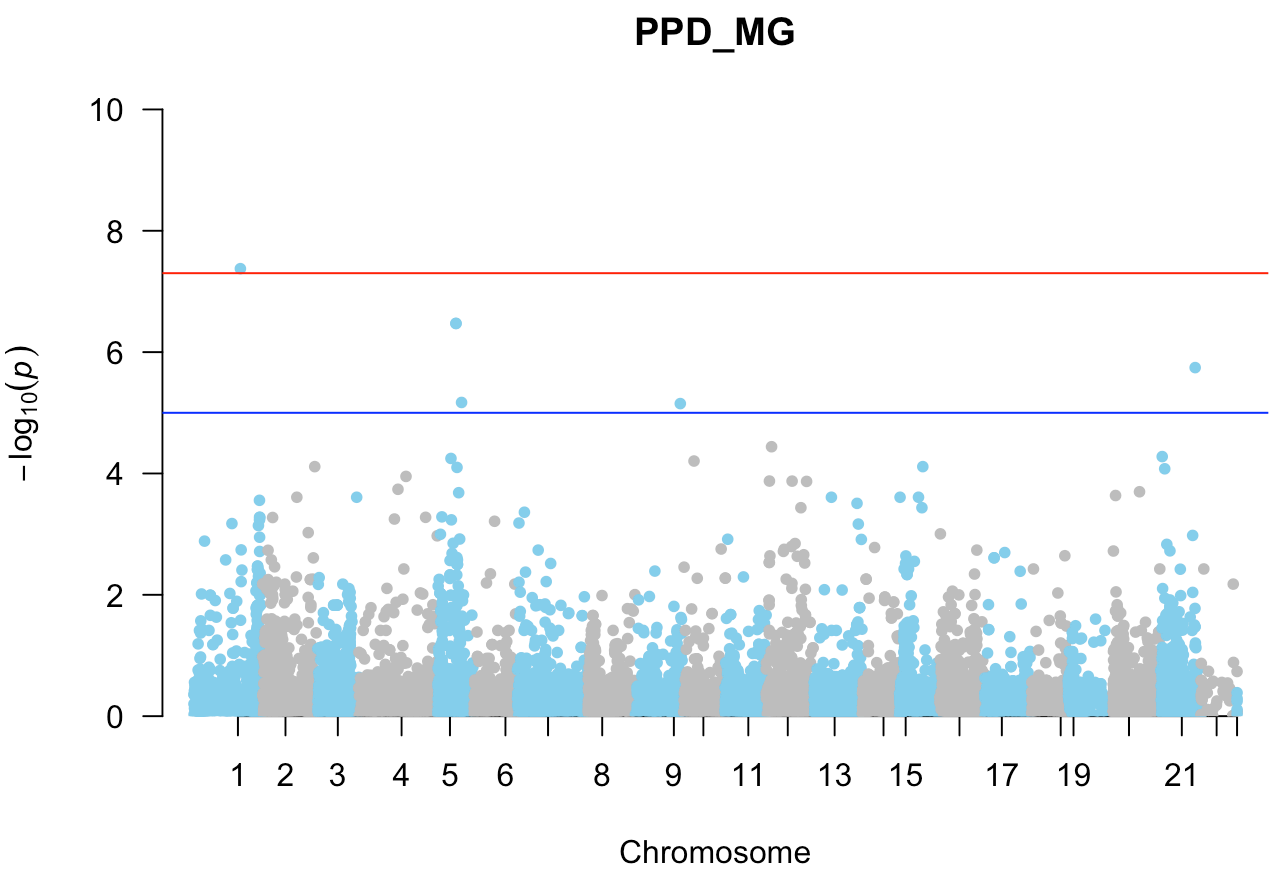

```
manhattan(maxHD, main = "Max HD", chr = "chr", bp = "pos", p = "p", snp = "ID", ylim = c
(0,
    10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # no GWS
```




```
manhattan(maxJP, main = "Max JP", chr = "chr", bp = "pos", p = "p", snp = "ID", ylim = c
(0,
    10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # no GWS
```



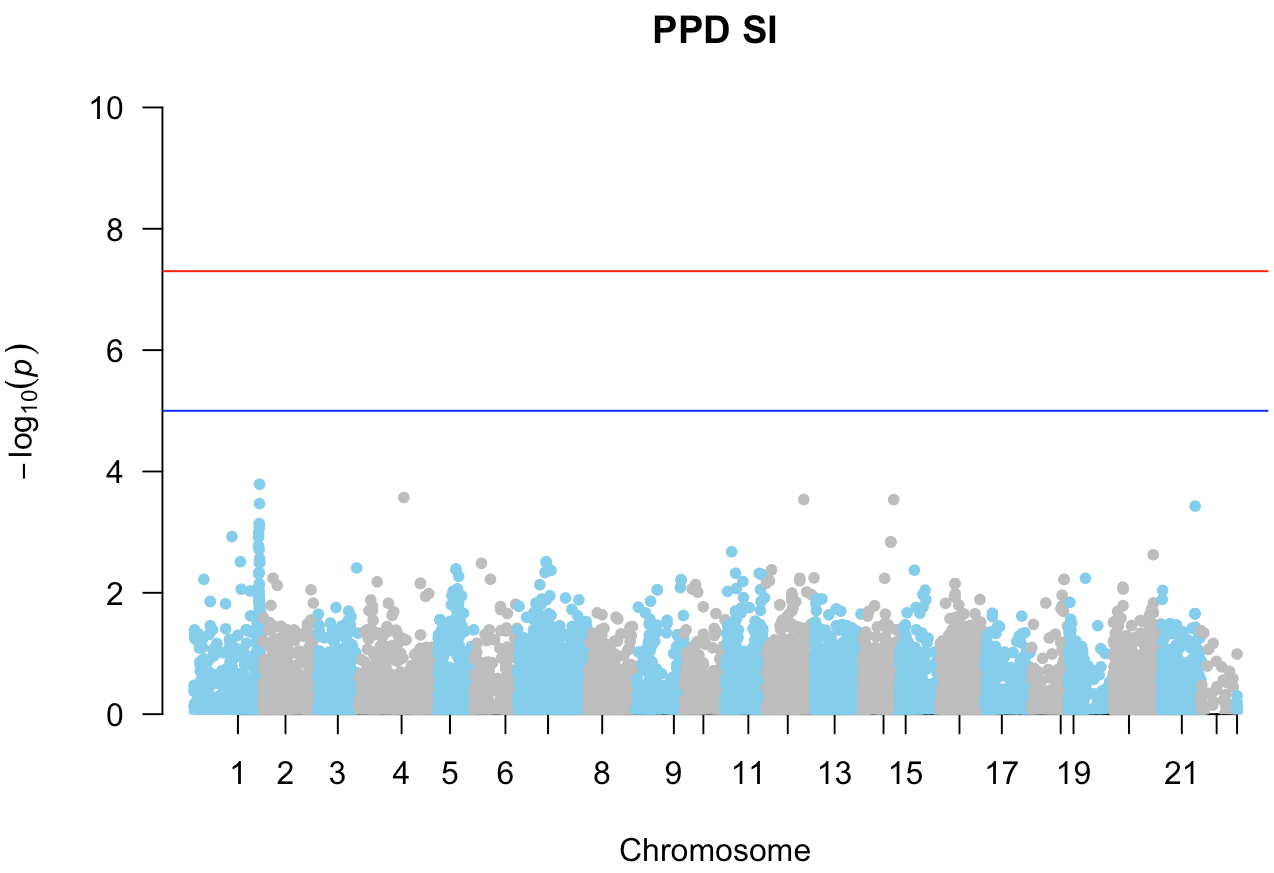

```
manhattan(PPD_MG, main = "PPD_MG", chr = "chr", bp = "pos", p = "p", snp = "ID",
    ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # GWS
```



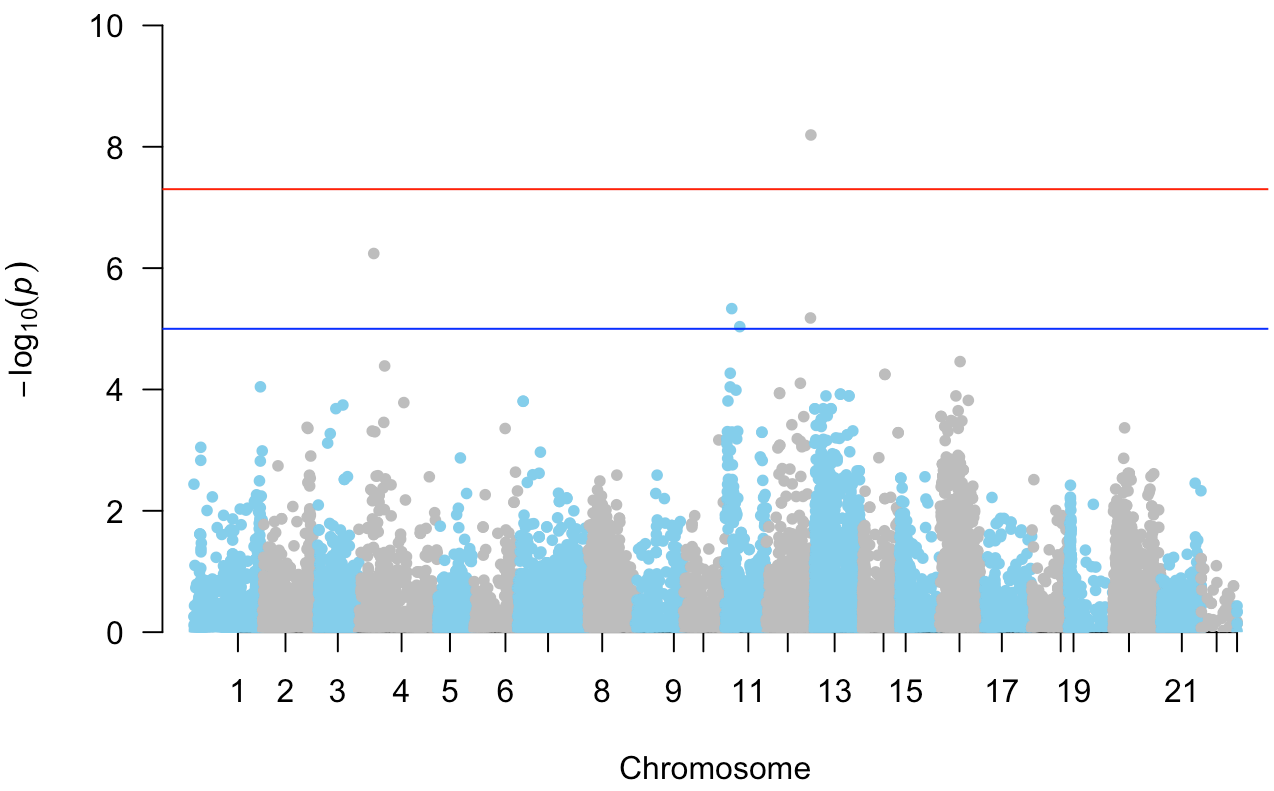

```
ppdmg_sig <- PPD_MG %>%
    filter(p < threshold)

PPD_MG %>%
    filter(p < threshold) %>%
    kbl() %>%
    kable_minimal()
```

chr	pos	ID	p
1	20019683	1:20019683	0

```
manhattan(PPD_SI, main = "PPD SI", chr = "chr", bp = "pos", p = "p", snp = "ID",
    ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # no GWS
```



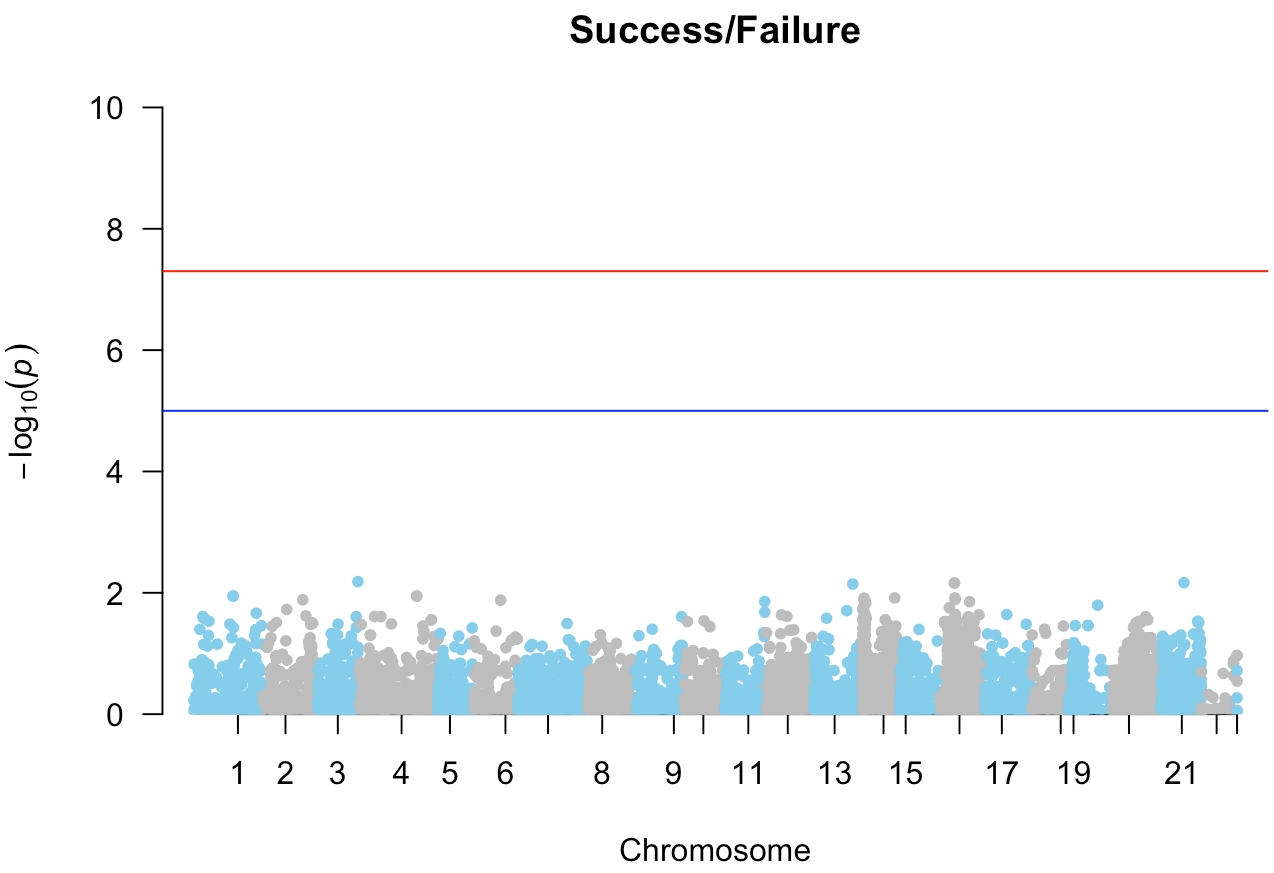

```
manhattan(ramSpeed, main = "Ram Speed", chr = "chr", bp = "pos", p = "p", snp = "ID",
    ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # GWS
```



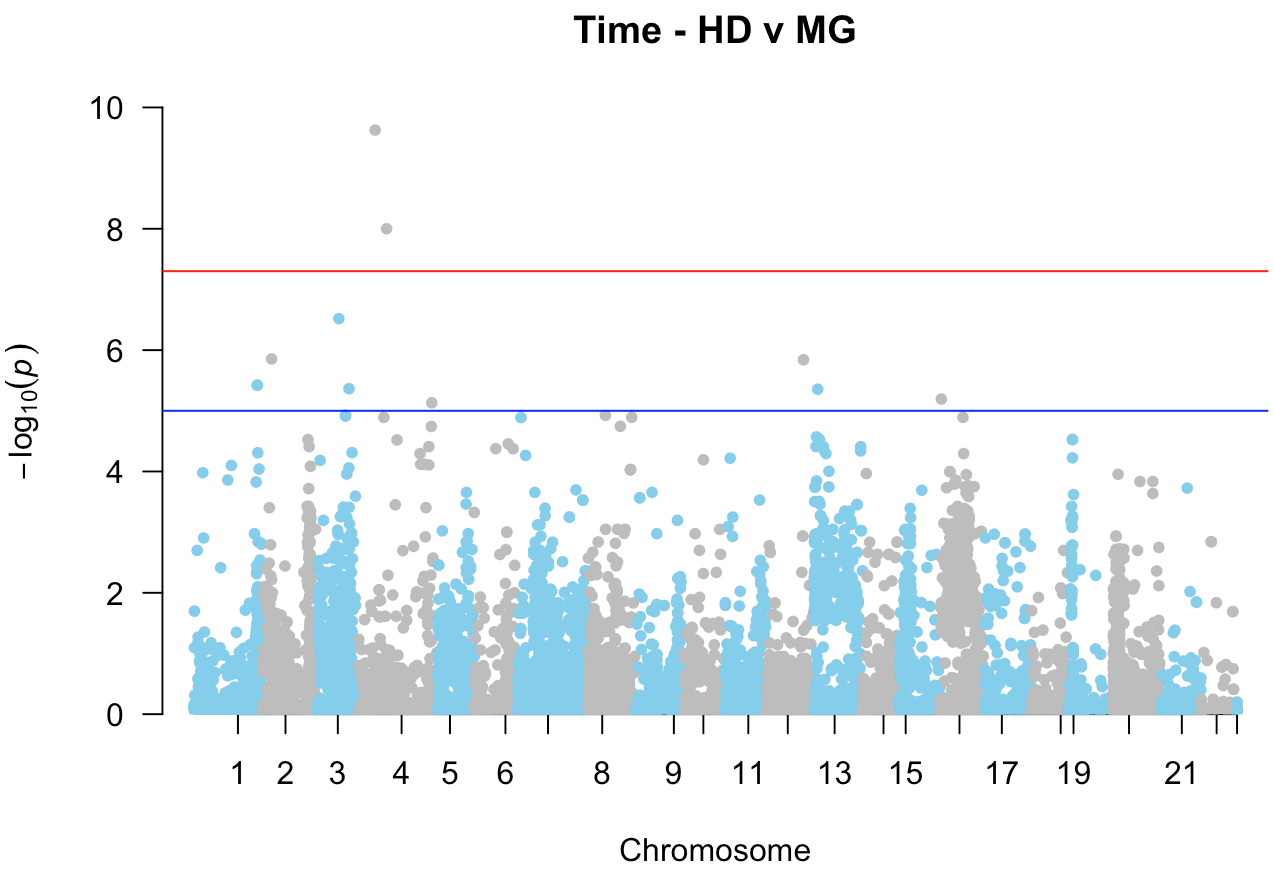

```
rs_sig <- ramSpeed %>%
    filter(p < threshold)

ramSpeed %>%
    filter(p < threshold) %>%
    kbl() %>%
    kable_minimal()
```

chr	pos	ID	p
12	19163490	12:19163490	0

```
manhattan(sf, main = "Success/Failure", chr = "chr", bp = "pos", p = "p", snp = "ID",
    ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # no GWS
```



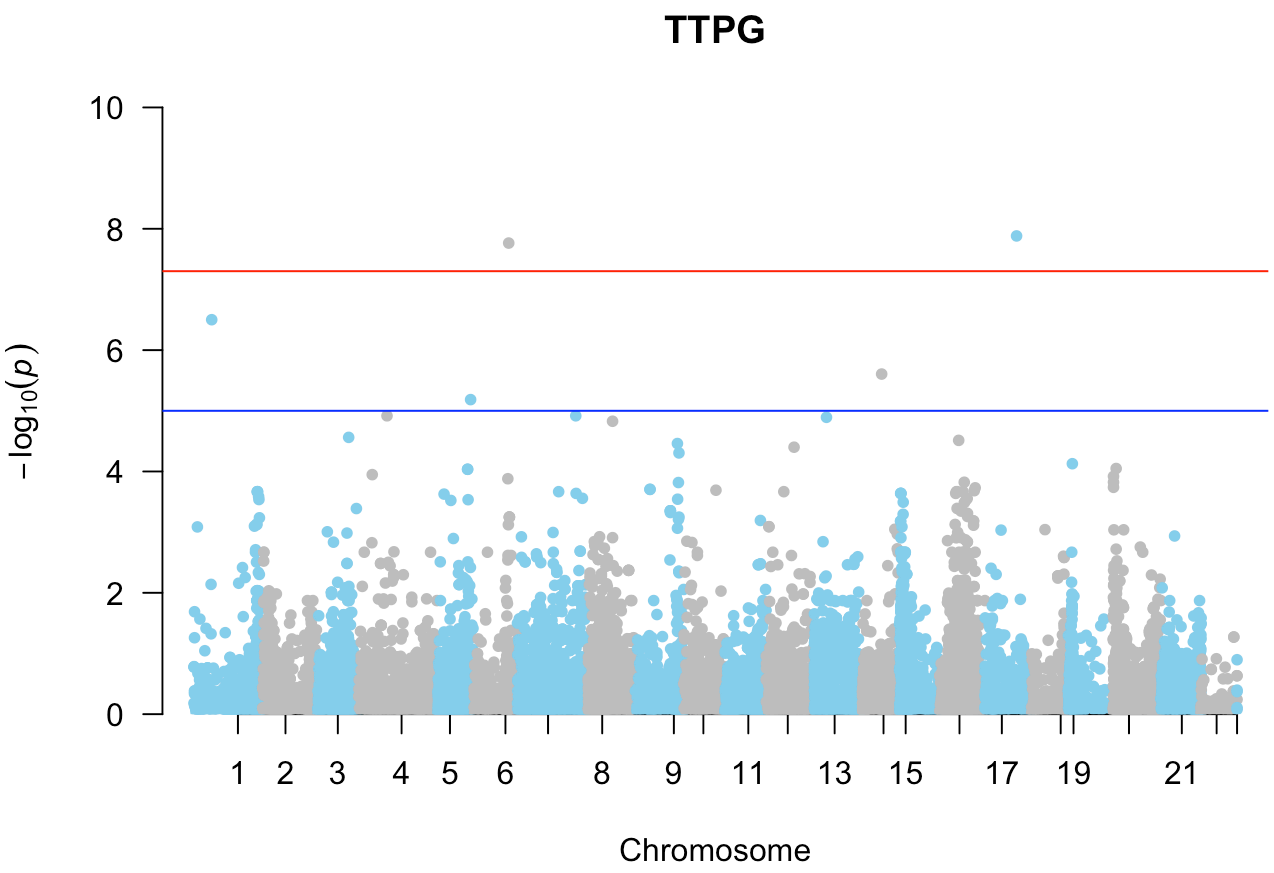

```
manhattan(time_HDvMG, main = "Time - HD v MG", chr = "chr", bp = "pos", p = "p",
    snp = "ID", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))  # GWS
```




```
thdvmg_sig <- time_HDvMG %>%
    filter(p < threshold)

time_HDvMG %>%
    filter(p < threshold) %>%
    kbl() %>%
    kable_minimal()
```

chr	pos	ID	p
4	7030660	4:7030660	0
4	11954636	4:11954636	0

```
manhattan(ttpg, main = "TTPG", chr = "chr", bp = "pos", p = "p", snp = "ID", ylim = c(0,
    10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue", "grey"))  # GWS
```




```
ttpg_sig <- ttpg %>%
    filter(p < threshold)

ttpg %>%
    filter(p < threshold) %>%
    kbl() %>%
    kable_minimal()
```

chr	pos	ID	p
6	14989382	6:14989382	0
17	13604080	17:13604080	0

Linkage Disequilibrium Analysis in SNPs with genome-wide significance

Read `.pvar` files in and save as new objects

```
file_paths_pvar <- list.files(file_dir, pattern = "gasAcu.plink.*\\.pvar$", full.names =
TRUE)

for (file in file_paths_pvar) {

    parts <- str_split(basename(file), "\\.", simplify = TRUE)
    name_pvar <- paste0(parts[, 3], ".pvar")

    data_pvar <- read.table(file, header = F, col.names = c("CHR", "POS", "ID", "REF",
        "ALT", "QUAL", "INFO")) %>%
        filter((CHR <= 22 | is.na(CHR))) %>%
        filter(CHR != 1.1 | is.na(CHR)) %>%
        filter(CHR != 21.1 | is.na(CHR)) %>%
        mutate(CHR = replace_na(CHR, 23)) %>%
        dplyr::select(-c(QUAL, INFO)) %>%
        rename(chr = CHR, pos = POS)

    assign(name_pvar, data_pvar, envir = .GlobalEnv)
}
```

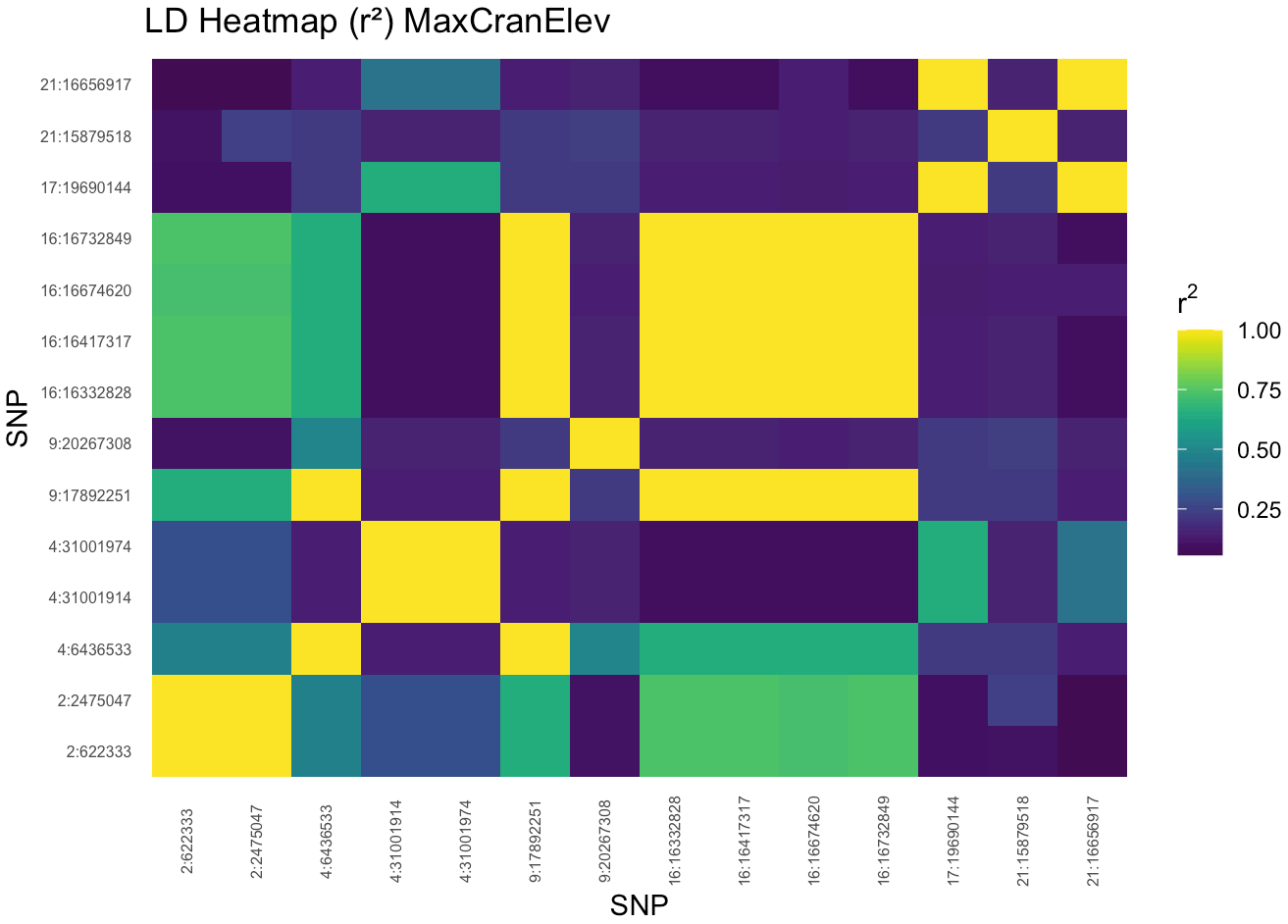
Associate significant SNPs with `.pvar` files, extract the significant SNPs, and write out

```
mce_sig_ann <- mce_sig %>%
    inner_join(maxCranElev.pvar, by = c("chr", "pos", "ID")) %>%
    dplyr::select(ID) %>%
    write_delim(., "gwas_results/mce_sig_snps.txt", delim = "\t")

md_sig_ann <- md_sig %>%
    inner_join(maxDecel.pvar, by = c("chr", "pos", "ID")) %>%
    dplyr::select(ID) %>%
    write_delim(., "gwas_results/md_sig_snps.txt", delim = "\t")

ppdmg_sig_ann <- ppdmg_sig %>%
    inner_join(PPD_MG.pvar, by = c("chr", "pos", "ID")) %>%
    dplyr::select(ID) %>%
    write_delim(., "gwas_results/ppdmg_sig_snps.txt", delim = "\t")

rs_sig_ann <- rs_sig %>%
    inner_join(ramSpeed.pvar, by = c("chr", "pos", "ID")) %>%
    dplyr::select(ID) %>%
    write_delim(., "gwas_results/rs_sig_snps.txt", delim = "\t")

thdvmg_sig_ann <- thdvmg_sig %>%
    inner_join(time_HDvMG.pvar, by = c("chr", "pos", "ID")) %>%
    dplyr::select(ID) %>%
    write_delim(., "gwas_results/thdvmg_sig_snps.txt", delim = "\t")

ttpg_sig_ann <- ttpg_sig %>%
    inner_join(ttpg.pvar, by = c("chr", "pos", "ID")) %>%
    dplyr::select(ID) %>%
    write_delim(., "gwas_results/ttpg_sig_snps.txt", delim = "\t")
```

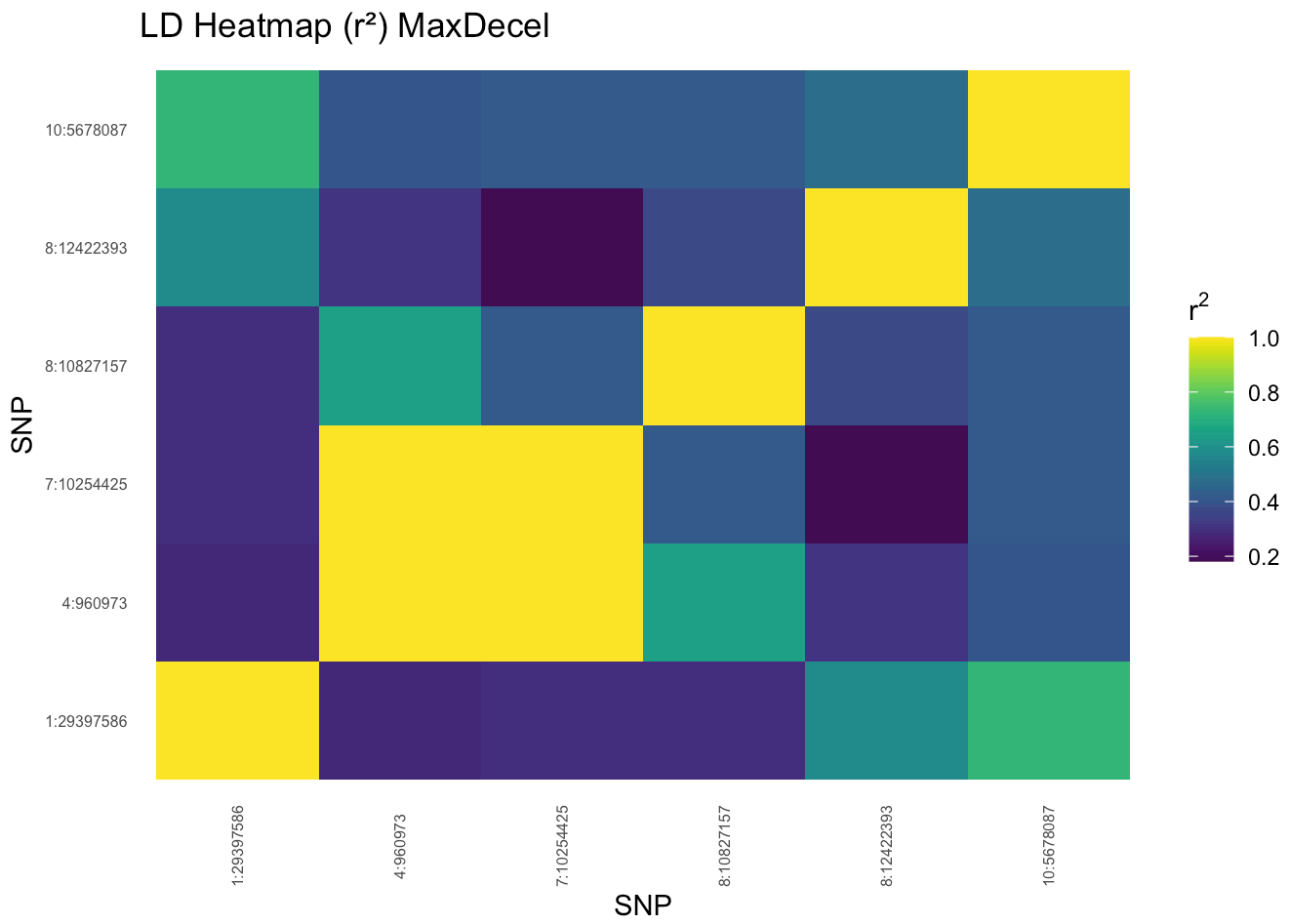
bash interlude to produce LD matrices

Visualize the phenotypes with mulitple significant SNPs as LD heatmaps

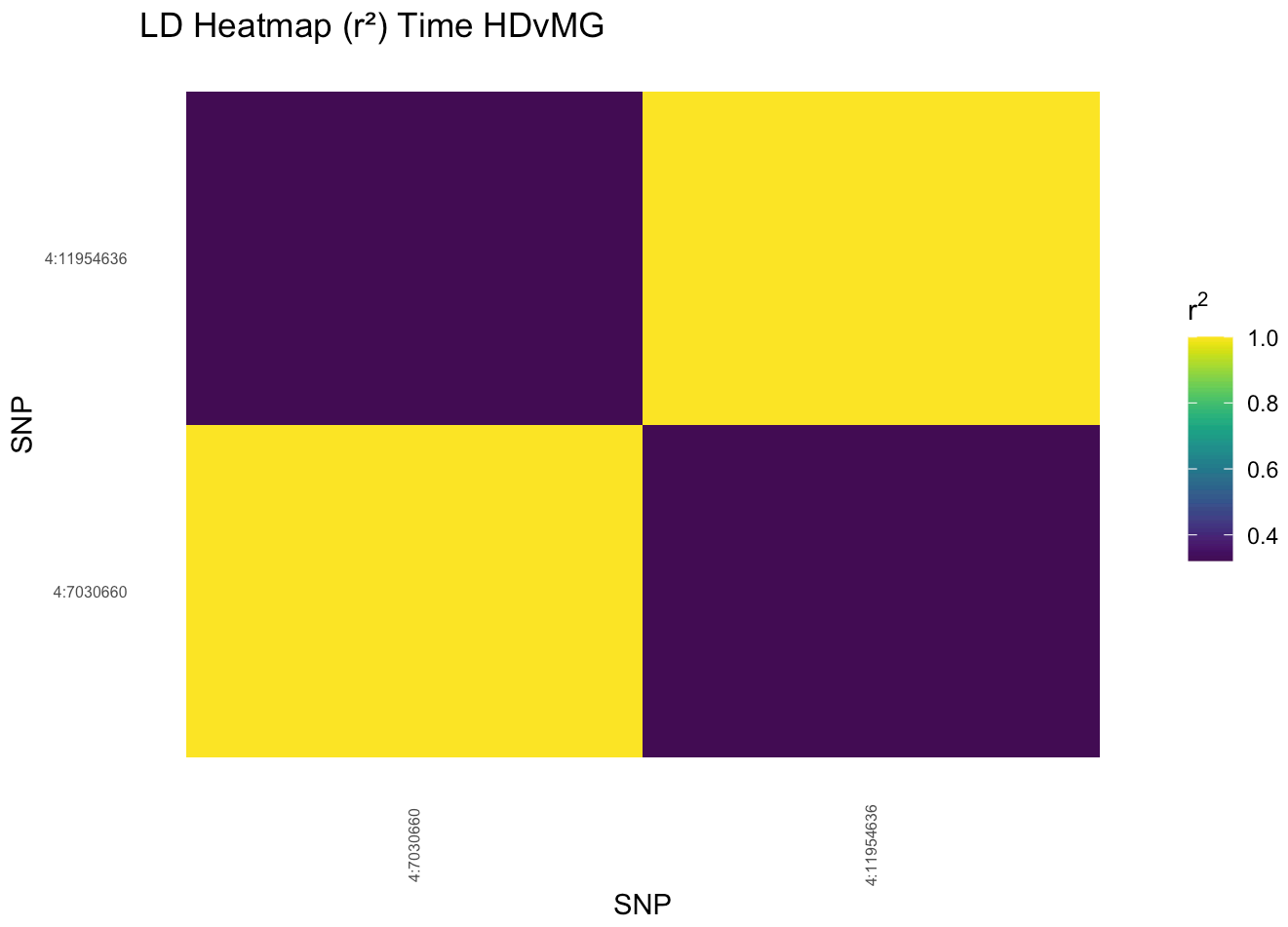
```
mce_ld_mat <- read.table("gwas_results/mce_ld_matrix.ld", header = F)
rownames(mce_ld_mat) <- mce_sig_ann$ID
colnames(mce_ld_mat) <- mce_sig_ann$ID

mce_ld_long <- melt(as.matrix(mce_ld_mat), varnames = c("SNP_A", "SNP_B"), value.name =
"R2")

ggplot(mce_ld_long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() + scale_fill_vir
idis(option = "viridis") +
    theme_minimal() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size =
6),
    axis.text.y = element_text(size = 6), panel.grid = element_blank()) + labs(title =
"LD Heatmap (r²) MaxCranElev",
    x = "SNP", y = "SNP", fill = expression(r^2))
```



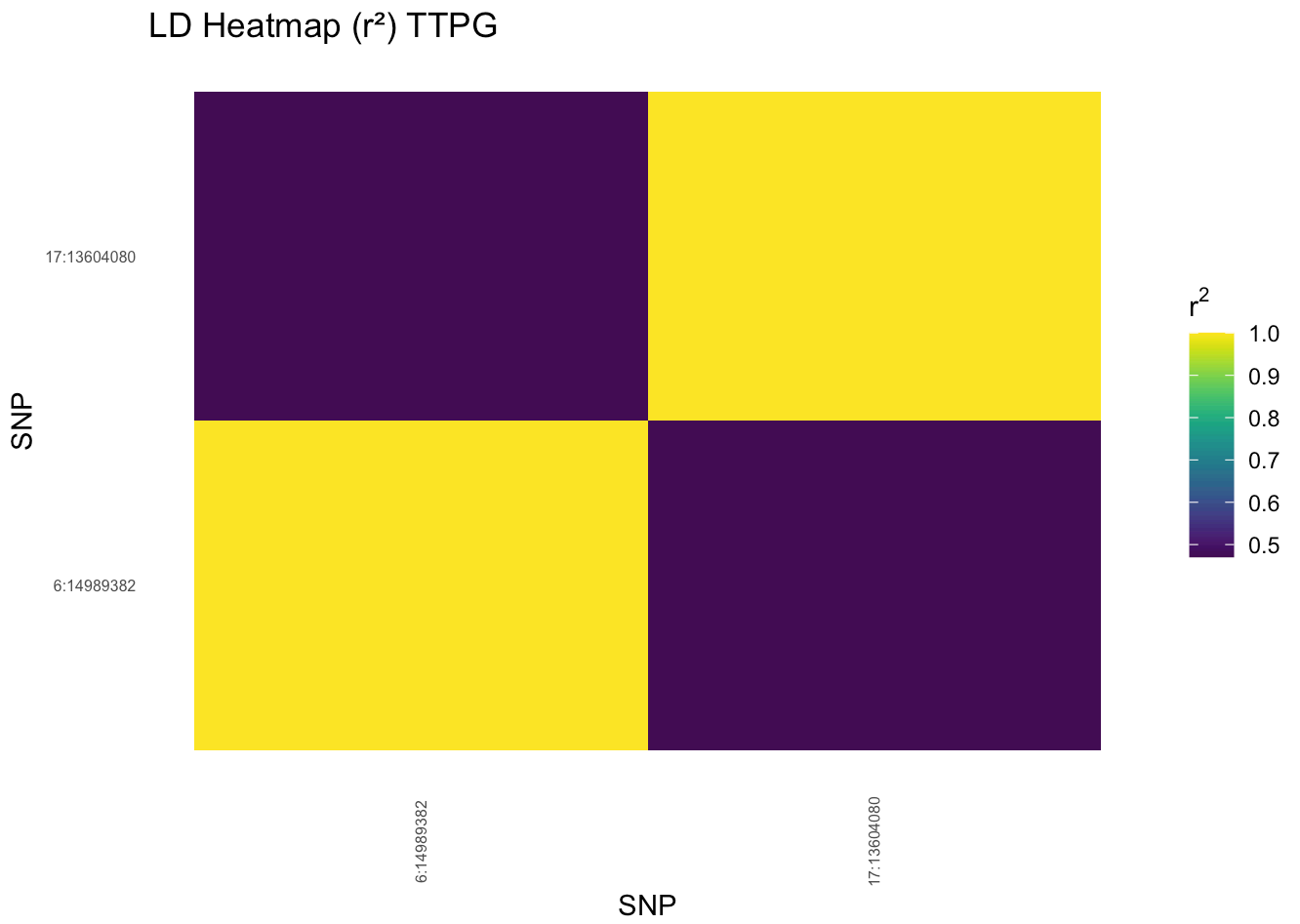
```
md_ld_mat <- read.table("gwas_results/md_ld_matrix.ld", header = F)
rownames(md_ld_mat) <- md_sig_ann$ID
colnames(md_ld_mat) <- md_sig_ann$ID

md_ld_long <- melt(as.matrix(md_ld_mat), varnames = c("SNP_A", "SNP_B"), value.name = "R
2")

ggplot(md_ld_long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() + scale_fill_viri
dis(option = "viridis") +
    theme_minimal() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size =
6),
    axis.text.y = element_text(size = 6), panel.grid = element_blank()) + labs(title =
"LD Heatmap (r²) MaxDecel",
    x = "SNP", y = "SNP", fill = expression(r^2))
```



```
thdvmg_ld_mat <- read.table("gwas_results/thdvmg_ld_matrix.ld", header = F)
rownames(thdvmg_ld_mat) <- thdvmg_sig_ann$ID
colnames(thdvmg_ld_mat) <- thdvmg_sig_ann$ID

thdvmg_ld_long <- melt(as.matrix(thdvmg_ld_mat), varnames = c("SNP_A", "SNP_B"),
    value.name = "R2")

ggplot(thdvmg_ld_long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() + scale_fill_
viridis(option = "viridis") +
    theme_minimal() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size =
6),
    axis.text.y = element_text(size = 6), panel.grid = element_blank()) + labs(title =
"LD Heatmap (r²) Time HDvMG",
    x = "SNP", y = "SNP", fill = expression(r^2))
```



```
ttpg_ld_mat <- read.table("gwas_results/ttpg_ld_matrix.ld", header = F)
rownames(ttpg_ld_mat) <- ttpg_sig_ann$ID
colnames(ttpg_ld_mat) <- ttpg_sig_ann$ID

ttpg_ld_long <- melt(as.matrix(ttpg_ld_mat), varnames = c("SNP_A", "SNP_B"), value.name
= "R2")

ggplot(ttpg_ld_long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() + scale_fill_vi
ridis(option = "viridis") +
    theme_minimal() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size =
6),
    axis.text.y = element_text(size = 6), panel.grid = element_blank()) + labs(title =
"LD Heatmap (r²) TTPG",
    x = "SNP", y = "SNP", fill = expression(r^2))
```



Make .BED for gene association

```
mce_bed <- mce_sig %>%
    inner_join(maxCranElev.pvar, by = c("chr", "pos", "ID")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "mce") %>%
    select(chr, start, end, ID, pheno)

md_bed <- md_sig %>%
    inner_join(maxDecel.pvar, by = c("chr", "pos", "ID")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "md") %>%
    select(chr, start, end, ID, pheno)

ppdmg_bed <- ppdmg_sig %>%
    inner_join(PPD_MG.pvar, by = c("chr", "pos", "ID")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "ppdmg") %>%
    select(chr, start, end, ID, pheno)

rs_bed <- rs_sig %>%
    inner_join(ramSpeed.pvar, by = c("chr", "pos", "ID")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "rs") %>%
    select(chr, start, end, ID, pheno)

thdvmg_bed <- thdvmg_sig %>%
    inner_join(time_HDvMG.pvar, by = c("chr", "pos", "ID")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "thdvmg") %>%
    select(chr, start, end, ID, pheno)

ttpg_bed <- ttpg_sig %>%
    inner_join(ttpg.pvar, by = c("chr", "pos", "ID")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "ttpg") %>%
    select(chr, start, end, ID, pheno)

clean_bed <- bind_rows(mce_bed, md_bed, ppdmg_bed, rs_bed, thdvmg_bed, ttpg_bed)
write_delim(clean_bed, "geneAssoc/cleanGenes.bed", delim = "\t", col_names = F)
```

bash interlude to produce BED

Create gene lists for associated SNPs

```
genes <- read.table("geneAssoc/gasAcu.genes.clean.bed", col.names = c("chr", "start", "e
nd", "genes", "snp", "pheno")) %>%
  arrange(pheno, chr, start, end) %>%
  select(snp, pheno, genes)

gene_split <- as_tibble(str_split(genes$genes, ",", n = Inf, simplify = T))

gene_clean <- genes %>%
  bind_cols(., gene_split) %>%
  filter(snp != "9:20267308") %>% # gets rid of one leftover LOC*
  select(-c(genes))

write_delim(gene_clean, "geneAssoc/gasAcu.final.cleanGenes.bed", delim = "\t")
write_delim(gene_clean, "geneAssoc/gasAcu.final.cleanGenes.txt", delim = "\t")
write_delim(gene_clean, "geneAssoc/gasAcu.final.cleanGenes.csv", delim = ",")
```