Genomics of Kinematics & Morphology in Threespine Stickleback

Dr. Sara J. Smith

2025-08-15

Contents

Kinematic traits 7
Morphological Traits 10
Visualizations - Manhattan plots (all traits) 12
Significant SNPs 52
Linkage Disequilibrium Analysis in SNPs with genome-wide significance 53
SNP Gene Association 60

HPC Environments

```
conda create -n ncbi -c conda-forge ncbi-datasets-cli

conda create -n fastq2vcf -c bioconda stacks fastqc multiqc bwa bcftools samtools fastp sambamba java-j
conda activate fastq2vcf
conda install -c bioconda openssl=1.0 # for bcftools issue

conda create -n vcfFilt -c bioconda plink vcftools htslib bcftools rename

conda create -n plink2 -c bioconda plink2

conda create -n bedtools -c bioconda -c conda-forge bedtools bedops
```

Reference Genome download

```
# in projects/def-sjsmith/sjsmith/stickles_ucr/reference
conda activate ncbi

datasets download genome accession GCF_016920845.1 --include gff3,genome,seq-report
unzip ncbi_dataset.zip
mv ncbi_dataset/data/* .
rm -r *.zip ncbi_dataset *.json*
mv GCF_016920845.1/genomic.gff GCF_016920845.1/gasAcu.gff

conda deactivate
```

```
#conda create -n stacks -c bioconda stacks fastqc multiqc bwa samtools bcftools (also outlined in 00_se
conda activate stacks

bwa index GCF_016920845.1_GAculeatus_UGA_version5_genomic.fna
```

fastq2vcf

```
# from /home/sjsmith/projects/def-sjsmith/sjsmith/stickles_ucr/seq_data/

conda activate fastq2vcf
module load samtools # conda conflicts

## Process FASTQs with STACKS
mkdir 01_demulti

process_radtags -1 ../00_raw_fastq/NS.LH00147_0009.006.B723---B503.THigham_20230515_plate1_R1.fastq.gz -
# 178395276 total sequences
# 82910 ambiguous barcode drops (0.0%)
# 0 low quality read drops (0.0%)
# 1383657 ambiguous RAD-Tag drops (0.8%)
# 176928709 retained reads (99.2%)

rm *rem*

# QC
mkdir 01_demulti/fastqc
for file in 01_demulti/*.fq
do
  fastqc $file -o 01_demulti/fastqc
done
multiqc 01_demulti/fastqc

mv multiqc_data/ multiqc_data_demulti/
mv multiqc_report.html multiqc_report_demulti.html

## Trim adapters with FastP
cd 01_demulti
ls *.1.fq | sed '/\.1\.fq/s///' > samples
mkdir -p trimmed/fastP_out

while read file
do
  fastp --in1 $file.1.fq --in2 $file.2.fq --out1 trimmed/$file.R1.fq --out2 trimmed/$file.R2.fq -q 15 -u
done < samples

# QC
mkdir -p trimmed/fastqc
cd ..

for file in 01_demulti/trimmed/*.fq
do
```

```
  fastqc $file -o 01_demulti/trimmed/fastqc
done
multiqc 01_demulti/trimmed/fastqc

mv multiqc_data/ multiqc_data_filtered/
mv multiqc_report.html multiqc_report_filtered.html

## Align reads to reference genome with BWA
mkdir 02_align

while read file
do
  bwa mem -O 5 -B 3 -a -M -t 16 ../reference/GCF_016920845.1/GCF_016920845.1_GAculeatus_UGA_version5_ge
done < 01_demulti/samples

# QC
cd 02_align
ls *.sam | sed '/\.sam/s///' > samples

while read file
do
  samtools view -Sbt ../../reference/GCF_016920845.1/GCF_016920845.1_GAculeatus_UGA_version5_genomic.fn
done < samples

## Convert SAM to BAM, sort, & index

while read file
do
  samtools view -q 20 -b -S $file.sam > $file.bam
  samtools sort $file.bam -o $file.sort.bam
  samtools index $file.sort.bam $file.sort.idx
done < samples

## Mark duplicates
while read file
do
  sambamba markdup $file.sort.bam $file.markdup.bam
done < samples

# QC & filtering
while read file
do
  samtools flagstat $file.markdup.bam > $file.markdup.aln.stat
done < samples

## Call variants & generate VCF with BCFtools
ls *.markdup.bam > bamList
cd ..
mkdir 03_vcf
cd 02_align

bcftools mpileup -C 50 -E -I --max-depth 8000 -f ../../reference/GCF_016920845.1/GCF_016920845.1_GAcule
bcftools call -v -c -f GQ 03_vcf/stickles_ucr.bcf > 03_vcf/stickles_ucr.vcf
```

QC: Duplicates

```
# in /home/sjsmith/projects/def-sjsmith/sjsmith/stickles_ucr/seq_data/02_align
## removing dups with sambamba then generating VCF to compare with impact of downstream removal of dups

conda activate fastq2vcf

mkdir dedup

## Remove duplicates
while read file
do
  sambamba markdup -r $file.sort.bam dedup/$file.dedup.bam
done < samples

# QC & filtering
while read file
do
  samtools flagstat dedup/$file.dedup.bam > dedup/$file.dedup.aln.stat
done < samples

## Call variants & generate VCF with BCFtools
ls dedup/*.dedup.bam > dedup/bamList
mkdir -p dedup/03_vcf

bcftools mpileup -C 50 -E -I --max-depth 8000 -f ../../reference/GCF_016920845.1/GCF_016920845.1_GAculea
bcftools call -v -c -f GQ dedup/03_vcf/stickles_ucr.dedup.bcf > dedup/03_vcf/stickles_ucr.dedup.vcf
```

QC: Filtering

```
## in /home/sjsmith/projects/def-sjsmith/sjsmith/stickles_ucr/
conda activate vcfFilt

mkdir vcf_filt
cp seq_data/02_align/dedup/03_vcf/stickles_ucr.dedup.vcf vcf_filt
cd vcf_filt

bgzip stickles_ucr.dedup.vcf -c > stickles_ucr.dedup.vcf.gz
tabix -p vcf stickles_ucr.dedup.vcf.gz

vcftools --gzvcf stickles_ucr.dedup.vcf.gz --remove-indels --max-missing 0.5 --mac 3 --minQ 30 --recode
# kept 470374/944649
vcftools --vcf stickles.filt1.recode.vcf --minDP 3 --recode --recode-INFO-all --out stickles.filt2
# kept 470374/470374
vcftools --vcf stickles.filt2.recode.vcf --missing-indv
# no one to remove
vcftools --vcf stickles.filt2.recode.vcf --remove-indels --min-alleles 2 --max-alleles 2 --max-missing
# kept 470374/470374
vcftools --vcf stickles.filt3.recode.vcf --maf 0.01 --minGQ 10 --max-meanDP 200 --recode --recode-INFO-
# kept 453918/470374 sites
vcftools --vcf stickles.filt4.recode.vcf --remove-indels --min-alleles 2 --max-alleles 2 --max-missing
```

```
# final step; kept 16466/470374 sites

mkdir int_filt
mv out.* stickles.filt* int_filt
tar zcvf int_filt.tar.gz int_filt/
rm -r int_filt/

cp stickles.filtered.recode.vcf ..
```

PLINK

```
# in /home/sjsmith/projects/def-sjsmith/sjsmith/stickles_ucr
conda activate vcfFilt

## chromosomes need to be converted from alpahnumeric to numeric values.
# create chromosome map
bcftools view -H stickles.filtered.recode.vcf | cut -f 1 | uniq | awk '{print $0}' > gasAcu.chrom.map.t

## edit in R (because I don't have the capacity to loop this rn) gasAcu.chrom.map.txt using sequence re

bcftools annotate stickles.filtered.recode.vcf --rename-chrs gasAcu.chromMap -o gasAcu.filter.chrRename

## remove individuals with no phenotypes (not F2)
vcftools --vcf gasAcu.filter.chrRename.vcf --remove-indv dedup/OBBB_1.dedup.bam --remove-indv dedup/OOB
# kept 16466 out of a possible 16466 Sites, 58/60 individuals

##### use the gasAcu.chrRename.final.recode.vcf for the S vs F GWAS

# there are two individuals who only had failed trials and therefore have to be removed
vcftools --vcf gasAcu.chrRename.final.recode.vcf --remove-indv dedup/BAM_19.dedup.bam --remove-indv dedu
# kept 56 out of 58 Individuals

##### use the gasAcu.chrRename.noFails.recode.vcf for the continuous variable GWAS

# remove individuals that don't have morph phenos
vcftools --vcf gasAcu.chrRename.final.recode.vcf --remove-indv dedup/BAM_18.dedup.bam --remove-indv dedu
# kept 56/58 invd, 16466/16466 sites

##### use the gasAcu.chrRename.morph.recode.vcf for the morphological variable GWAS

conda deactivate
conda activate plink2

#success-failure
plink2 --vcf gasAcu.chrRename.final.recode.vcf --make-pgen --allow-extra-chr --set-all-var-ids @:# --sn

# ppdmg
plink2 --vcf gasAcu.chrRename.final.recode.vcf --make-pgen --allow-extra-chr --set-all-var-ids @:# --sn

# continuous traits
while IFS= read -r file
```

```
do
  plink2 --vcf gasAcu.chrRename.noFails.recode.vcf --make-pgen --allow-extra-chr --set-all-var-ids @:# +
done < "cont.phenos"

# morphological traits
while read file
do
  plink2 --vcf gasAcu.chrRename.morph.recode.vcf --make-pgen --allow-extra-chr --set-all-var-ids @:# --s
done < morph.phenos

# convert plink 2.0 to plink 1.9 for LD analyses

while read file
do
  plink2 --pfile gasAcu.plink.$file --make-bed --allow-extra-chr --out gasAcu.plink19.$file
done < cont.phenos

plink2 --pfile gasAcu.plink.sf --make-bed --allow-extra-chr --out gasAcu.plink19.sf

plink2 --pfile gasAcu.plink.ppdmg --make-bed --allow-extra-chr --out gasAcu.plink19.ppdmg

while read file
do
  plink2 --pfile gasAcu.plink.$file --make-bed --allow-extra-chr --out gasAcu.plink19.$file
done < morph.phenos

mkdir gwas_results
mkdir -p gwas_results/morph

while IFS= read -r file
do
  mv gasAcu.plink.$file.* gwas_results/
done < "cont.phenos"

mv gasAcu.plink.sf* gwas_results/

mv gasAcu.plink.ppdmg* gwas_results/

mv gasAcu.plink* gwas_results/morph
```

R Libraries

```
# library(BiocManager) BiocManager::install('GENESIS', force = T)
library(GENESIS)
library(GWASTools)
library(SNPRelate)
library(tidyverse)
library(qqman)
library(topr)
library(reshape2)
library(viridis)
library(readxl)
```

Kinematic traits

Loading phenotypes

```
sf.pheno <- read_delim("~/Desktop/MRU_Faculty/Research/stickles_ucr/gwas_results/phenotypes_gwas/phenos
    rename(scanID = IID) %>%
    mutate(sf = if_else(sf == 1, 0, sf), sf = if_else(sf == 2, 1, sf))

ppdmg.pheno <- read_delim("~/Desktop/MRU_Faculty/Research/stickles_ucr/gwas_results/phenotypes_gwas/phen
    rename(scanID = IID)

cont.phenos <- read_delim("~/Desktop/MRU_Faculty/Research/stickles_ucr/gwas_results/phenotypes_gwas/phen
    rename(scanID = IID)
```

Creating scan annotation dfs

```
scanAnnot.sf <- ScanAnnotationDataFrame(sf.pheno)
scanAnnot.cont <- ScanAnnotationDataFrame(cont.phenos)
scanAnnot.ppdmg <- ScanAnnotationDataFrame(ppdmg.pheno)
```

Starting with continuous traits

```
files <- c("dist", "maxCranElev", "maxDecel", "maxGape", "maxHD", "maxJP", "ppdmg",
    "PPD_SI", "ramSpeed", "time_HDvMG", "time_maxDecelvMG", "ttpg")

# set up GDS function
make_gds <- function(pheno) {
    snpgdsBED2GDS(bed.fn = paste0("gwas_results/gwas_grm/gasAcu.plink19.", pheno,
        ".bed"), bim.fn = paste0("gwas_results/gwas_grm/gasAcu.plink19.", pheno,
        ".bim"), fam.fn = paste0("gwas_results/gwas_grm/gasAcu.plink19.", pheno,
        ".fam"), out.gdsfn = paste0("gwas_results/gwas_grm/", pheno, ".gds"), cvt.chr = "char")
}

# create GDS files
for (pheno in files) {
    make_gds(pheno)
}
```

```
# set up geno data function
create_geno <- function(pheno) {
    geno <- GdsGenotypeReader(filename = paste0("gwas_results/gwas_grm/", pheno,
        ".gds"))
    genoData <- GenotypeData(geno)
    assign(paste0("genoData.", pheno), genoData, envir = .GlobalEnv)
}

# create genotype data
for (pheno in files) {
```

```
    create_geno(pheno)
}
```

```
# set up KING matrix function
create_kin <- function(pheno) {
    gds <- snpgdsOpen(paste0("gwas_results/gwas_grm/", pheno, ".gds"), readonly = F,
        allow.duplicate = T)
    kin <- snpgdsIBDKING(gds)
    kin.mat <- kingToMatrix(kin)
    assign(paste0("kin.mat.", pheno), kin.mat, envir = .GlobalEnv)
    snpgdsClose(gds)
}

# create KING matrices
for (pheno in files) {
    create_kin(pheno)
}
```

```
# scanAnnot.cont names i.e., no s/f, no ppdmg
cont.files <- c("dist", "maxCranElev", "maxDecel", "maxGape", "maxHD", "maxJP", "PPD_SI",
    "ramSpeed", "time_HDvMG", "time_maxDecelvMG", "ttpg")

# create null models
for (pheno in cont.files) {
    kin_name <- paste0("kin.mat.", pheno)
    null_mod <- fitNullModel(scanAnnot.cont, outcome = pheno, cov.mat = get(kin_name),
        family = "gaussian")

    assign(paste0("null.mod.", pheno), null_mod, envir = .GlobalEnv)
}
```

Creating null models

```
for (pheno in cont.files) {
    geno_data <- get(paste0("genoData.", pheno))
    genoIterator <- GenotypeBlockIterator(geno_data, snpBlock = 10000)
    assoc <- assocTestSingle(genoIterator, null.model = get(paste0("null.mod.", pheno)),
        BPPARAM = BiocParallel::SerialParam())

    assign(paste0("assoc.", pheno), assoc, envir = .GlobalEnv)

    assoc_clean <- assoc %>%
        mutate(chr = if_else(chr == "U", "23", chr), chr = as.numeric(chr)) %>%
        rename(chrom = chr, pos = pos, p = Score.pval)

    assign(paste0("assoc.clean.", pheno), assoc_clean, envir = .GlobalEnv)
```

```
}
```

run GWAS analyses for continous phenotypes, clean up results, write out to env

```
## Using 1 CPU cores
## Using 1 CPU cores
## Using 1 CPU cores
## Using 1 CPU cores
## Using 1 CPU cores
## Using 1 CPU cores
## Using 1 CPU cores
## Using 1 CPU cores
## Using 1 CPU cores
## Using 1 CPU cores
```

PPDMG

```
snpgdsBED2GDS(bed.fn = "gwas_results/gwas_grm/gasAcu.plink19.ppdmg.bed", bim.fn = "gwas_results/gwas_grm
    fam.fn = "gwas_results/gwas_grm/gasAcu.plink19.ppdmg.fam", out.gdsfn = "gwas_results/gwas_grm/gasAcu
    cvt.chr = "char")

geno.ppdmg <- GdsGenotypeReader(filename = "gwas_results/gwas_grm/gasAcu.plink19.ppdmg.gds")
genoData.ppdmg <- GenotypeData(geno.ppdmg)

gds.ppdmg <- snpgdsOpen("gwas_results/gwas_grm/gasAcu.plink19.ppdmg.gds", readonly = F,
    allow.duplicate = T)

kin.ppdmg <- snpgdsIBDKING(gds.ppdmg)

kin.mat.ppdmg <- kingToMatrix(kin.ppdmg)

snpgdsClose(gds.ppdmg)

null.mod.ppdmg <- fitNullModel(scanAnnot.ppdmg, outcome = "PPD_MG", cov.mat = kin.mat.ppdmg,
    family = "gaussian")

genoIterator.ppdmg <- GenotypeBlockIterator(genoData.ppdmg, snpBlock = 10000)

assoc.ppdmg <- assocTestSingle(genoIterator.ppdmg, null.model = null.mod.ppdmg, BPPARAM = BiocParallel:

assoc.ppdmg.clean <- assoc.ppdmg %>%
    mutate(chr = if_else(chr == "U", "23", chr), chr = as.numeric(chr)) %>%
    rename(chrom = chr, pos = pos, p = Score.pval)
```

Strike Success/Fail

```
snpgdsBED2GDS(bed.fn = "gwas_results/gwas_grm/gasAcu.plink19.sf.bed", bim.fn = "gwas_results/gwas_grm/ga
    fam.fn = "gwas_results/gwas_grm/gasAcu.plink19.sf.fam", out.gdsfn = "gwas_results/gwas_grm/gasAcu.pl
```

```
    cvt.chr = "char")

geno.sf <- GdsGenotypeReader(filename = "gwas_results/gwas_grm/gasAcu.plink19.sf.gds")
genoData.sf <- GenotypeData(geno.sf)

gds.sf <- snpgdsOpen("gwas_results/gwas_grm/gasAcu.plink19.sf.gds", readonly = F,
    allow.duplicate = T)

kin.sf <- snpgdsIBDKING(gds.sf)

kin.mat.sf <- kingToMatrix(kin.sf)

snpgdsClose(gds.sf)

null.mod.sf <- fitNullModel(scanAnnot.sf, outcome = "sf", cov.mat = kin.mat.sf, family = "binomial")

genoIterator.sf <- GenotypeBlockIterator(genoData.sf, snpBlock = 10000)

assoc.sf <- assocTestSingle(genoIterator.sf, null.model = null.mod.sf, BPPARAM = BiocParallel::SerialPa

assoc.sf.clean <- assoc.sf %>%
    mutate(chr = if_else(chr == "U", "23", chr), chr = as.numeric(chr)) %>%
    rename(chrom = chr, pos = pos, p = Score.pval)
```

Morphological Traits

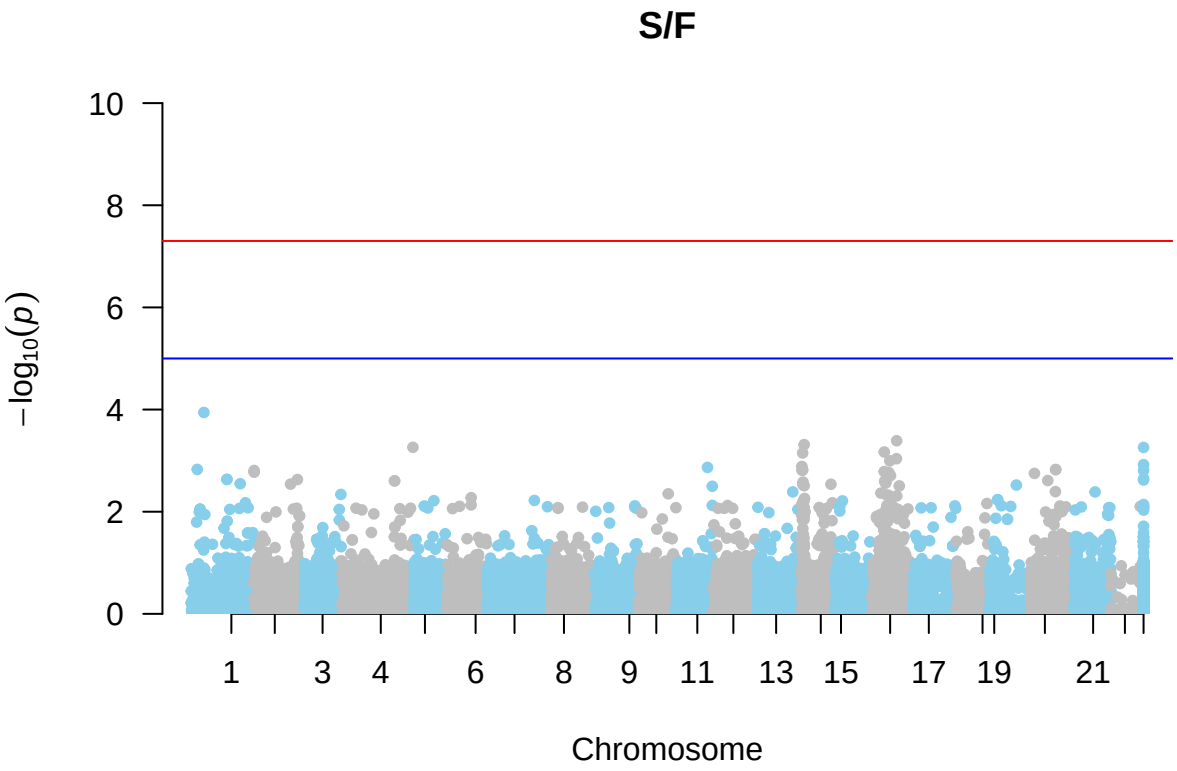
```
morph.phenos <- read_delim("~/Desktop/MRU_Faculty/Research/stickles_ucr/gwas_results/morphology/phenos/
    rename(scanID = IID)

scanAnnot.morph <- ScanAnnotationDataFrame(morph.phenos)

morph <- c("length", "height", "eye.dia", "caudal.area", "pec.length", "pec.area",
    "ray")

# make GDS function
make_gds <- function(pheno) {
    snpgdsBED2GDS(bed.fn = paste0("gwas_results/morphology/plink_out/gasAcu.plink19.",
        pheno, ".bed"), bim.fn = paste0("gwas_results/morphology/plink_out/gasAcu.plink19.",
        pheno, ".bim"), fam.fn = paste0("gwas_results/morphology/plink_out/gasAcu.plink19.",
        pheno, ".fam"), out.gdsfn = paste0("gwas_results/morphology/plink_out/",
        pheno, ".gds"), cvt.chr = "char")
}

for (pheno in morph) {
    make_gds(pheno)
}

# create KING matrices & geno data
create_geno <- function(pheno) {
    geno <- GdsGenotypeReader(filename = paste0("gwas_results/morphology/plink_out/",
        pheno, ".gds"))
```

```
    genoData <- GenotypeData(geno)
    assign(paste0("genoData.", pheno), genoData, envir = .GlobalEnv)
}

for (pheno in morph) {
    create_geno(pheno)
}

# create regularization function becuase otherwise nothing works (need positive
# definite matrix)
regularize_matrix <- function(K, lambda = 0.01) {
    (1 - lambda) * K + lambda * diag(nrow(K))
}

# create KING matrix
create_kin <- function(pheno) {
    gds <- snpgdsOpen(paste0("gwas_results/morphology/plink_out/", pheno, ".gds"),
        readonly = F, allow.duplicate = T)
    kin <- snpgdsIBDKING(gds)
    kin.mat <- kingToMatrix(kin)
    assign(paste0("kin.mat.", pheno), kin.mat, envir = .GlobalEnv)
    snpgdsClose(gds)
}

for (pheno in morph) {
    create_kin(pheno)
    kin.mat <- get(paste0("kin.mat.", pheno))
    kin.mat.reg <- regularize_matrix(kin.mat, lambda = 0.05)  # adjust lambda as needed
    assign(paste0("kin.mat.", pheno), kin.mat.reg, envir = .GlobalEnv)
}


# create null models
for (pheno in morph) {
    kin_name <- paste0("kin.mat.", pheno)
    null_mod <- fitNullModel(scanAnnot.morph, outcome = pheno, cov.mat = get(kin_name),
        family = "gaussian")

    assign(paste0("null.mod.", pheno), null_mod, envir = .GlobalEnv)
}

# run GWAS
for (pheno in morph) {
    geno_data <- get(paste0("genoData.", pheno))
    genoIterator <- GenotypeBlockIterator(geno_data, snpBlock = 10000)
    assoc <- assocTestSingle(genoIterator, null.model = get(paste0("null.mod.", pheno)),
        BPPARAM = BiocParallel::SerialParam())

    assign(paste0("assoc.", pheno), assoc, envir = .GlobalEnv)

    assoc_clean <- assoc %>%
        mutate(chr = if_else(chr == "U", "23", chr), chr = as.numeric(chr)) %>%
        rename(chrom = chr, pos = pos, p = Score.pval)
```

```
    assign(paste0("assoc.clean.", pheno), assoc_clean, envir = .GlobalEnv)


}
```

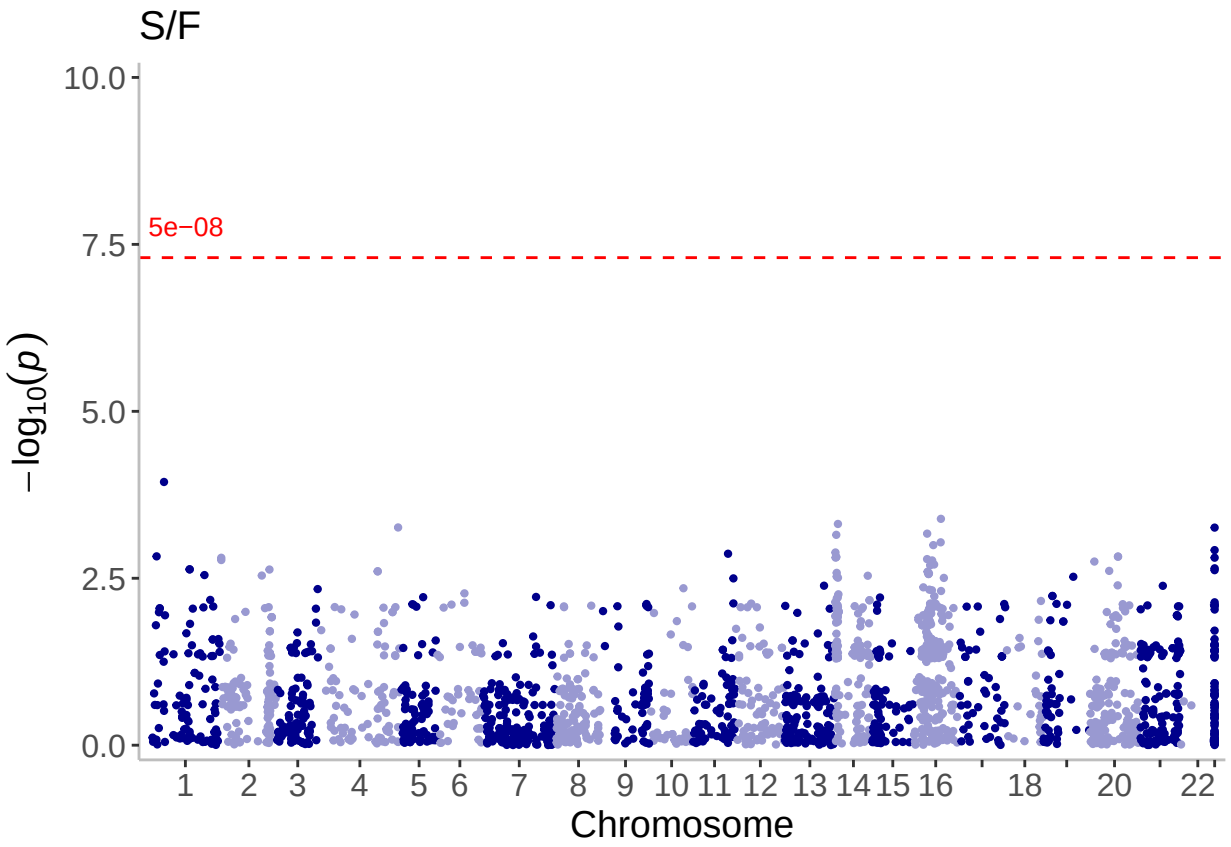
Visualizations - Manhattan plots (all traits)

```
# s/f - no GWS
qqman::manhattan(assoc.sf.clean, main = "S/F", chr = "chrom", bp = "pos", p = "p",
    snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```

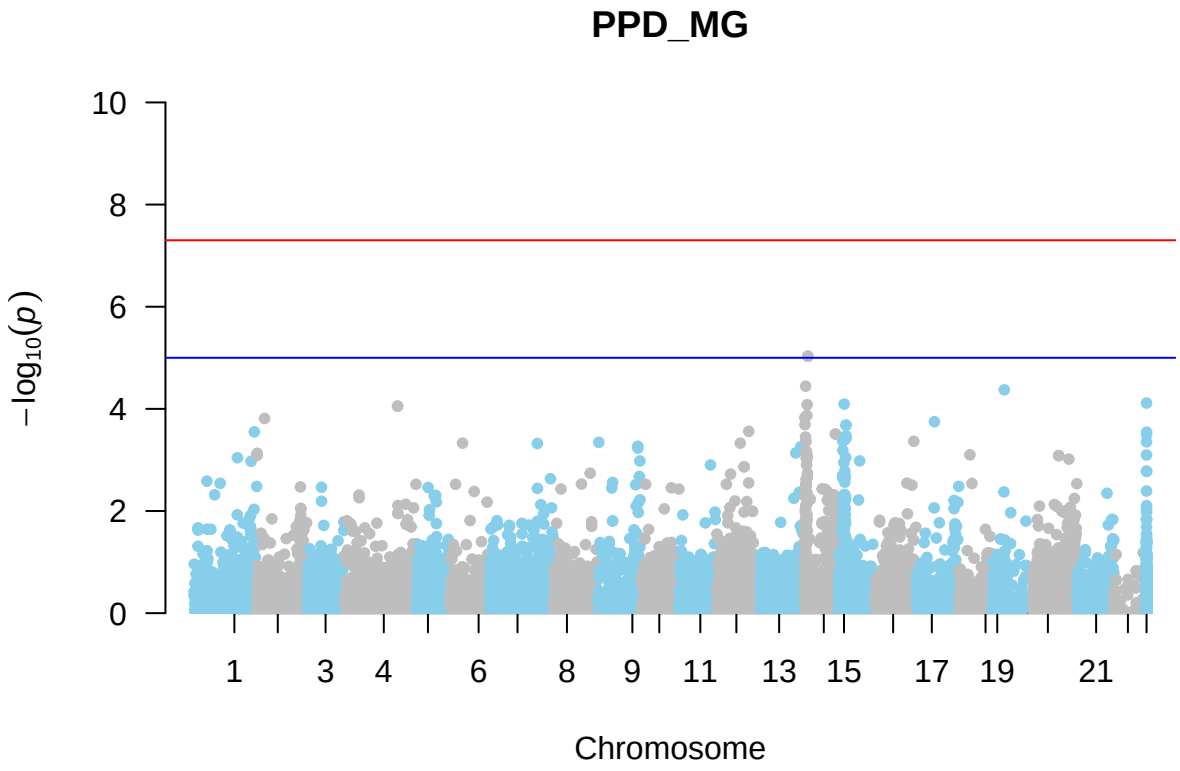



```
manhattan(assoc.sf.clean, title = "S/F", annotate = 1e-05, ymin = 0, ymax = 10)
```

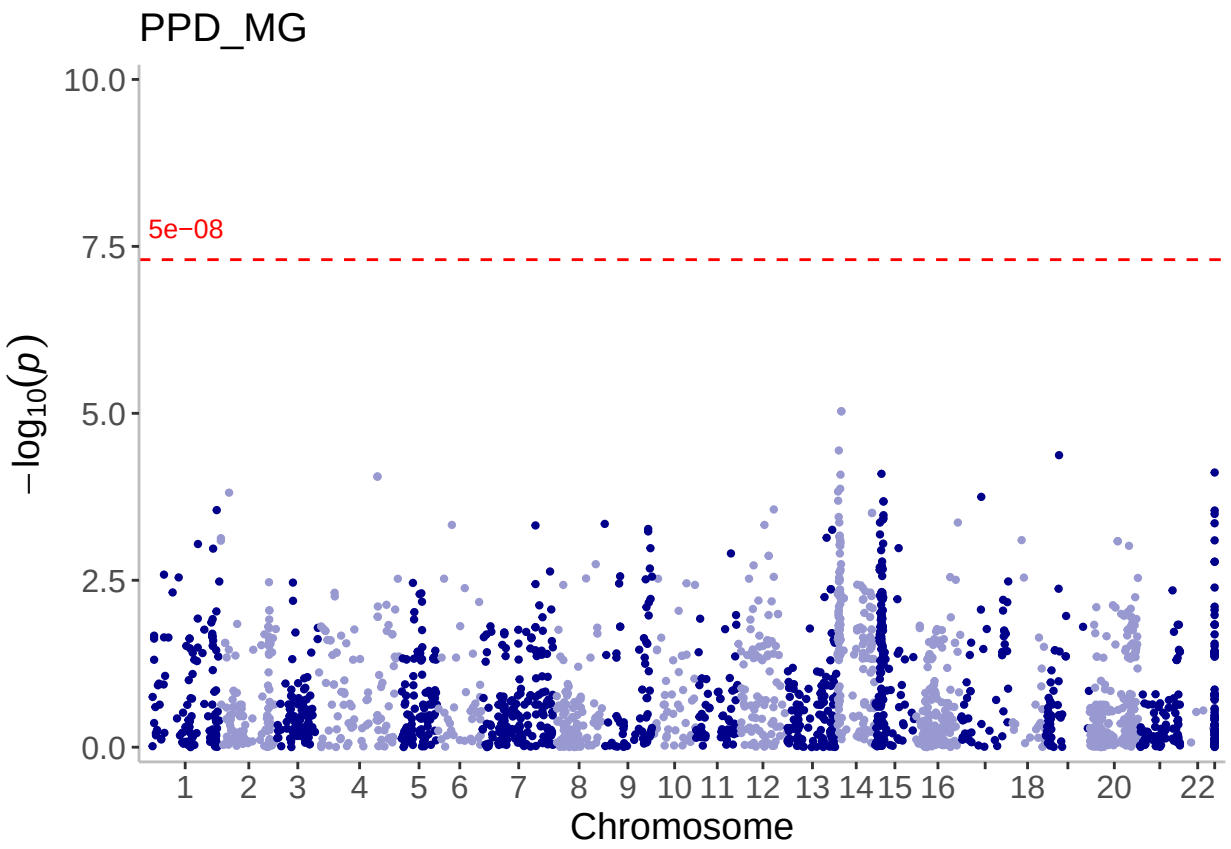
```
## [1] "There are no SNPs with p-values below 1e-05 in the input dataset. Use the [thresh] argument to l
```



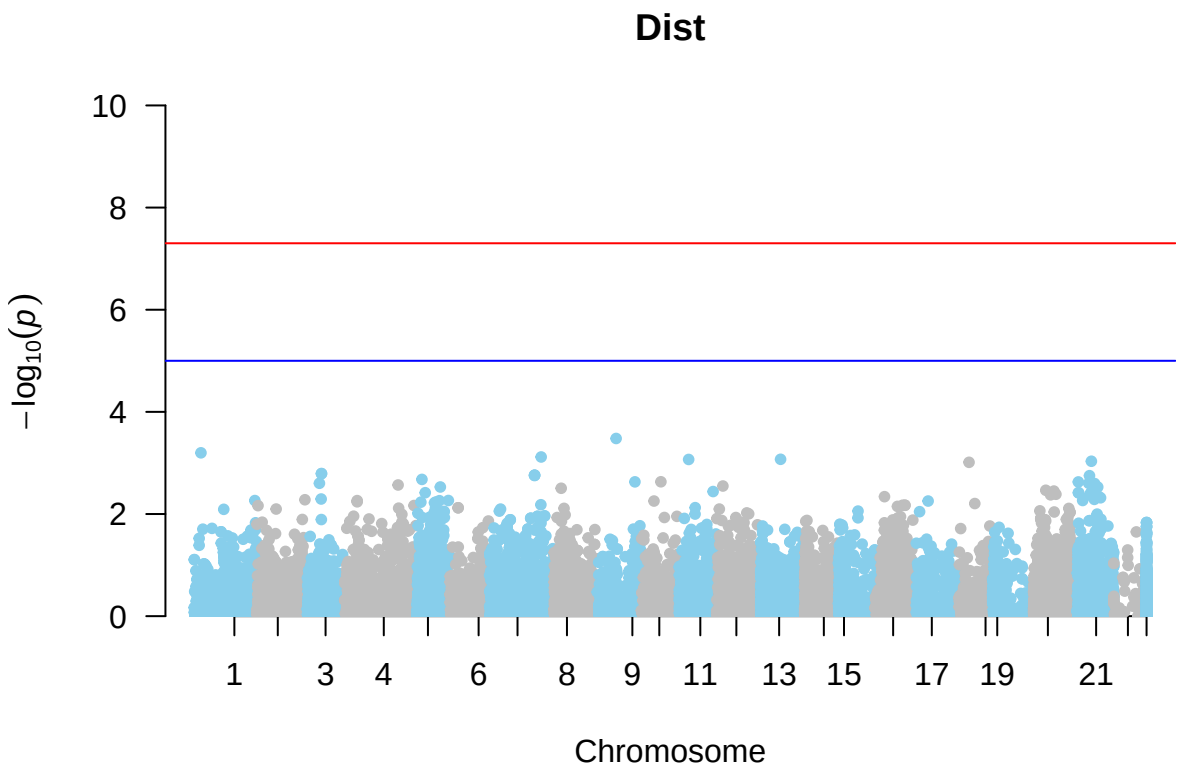

```
# ppdmg - no GWS
qqman::manhattan(assoc.ppdmg.clean, main = "PPD_MG", chr = "chrom", bp = "pos", p = "p",
    snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```




```
manhattan(assoc.ppdmg.clean, title = "PPD_MG", ymin = 0, ymax = 10)
```

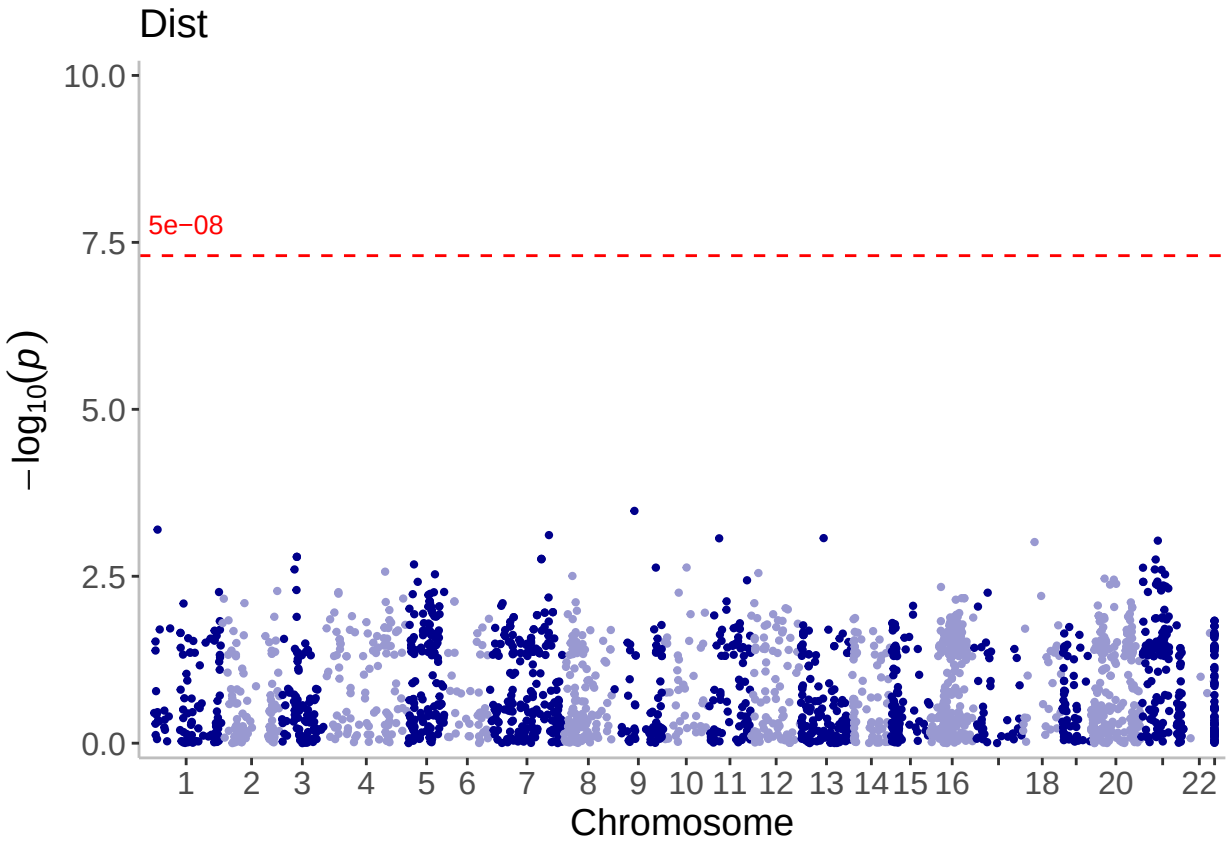



```
# dist - no GWS
qqman::manhattan(assoc.clean.dist, main = "Dist", chr = "chrom", bp = "pos", p = "p",
    snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```

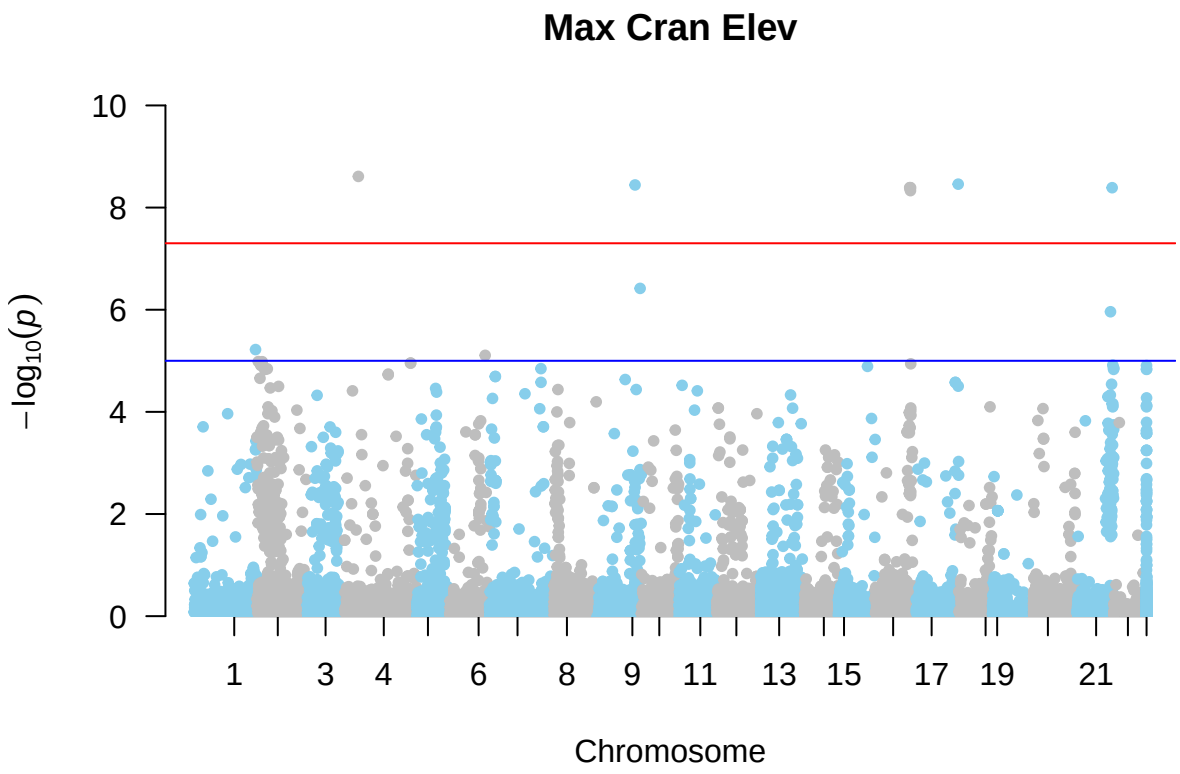



```
manhattan(assoc.clean.dist, title = "Dist", annotate = 1e-05, ymin = 0, ymax = 10)
```

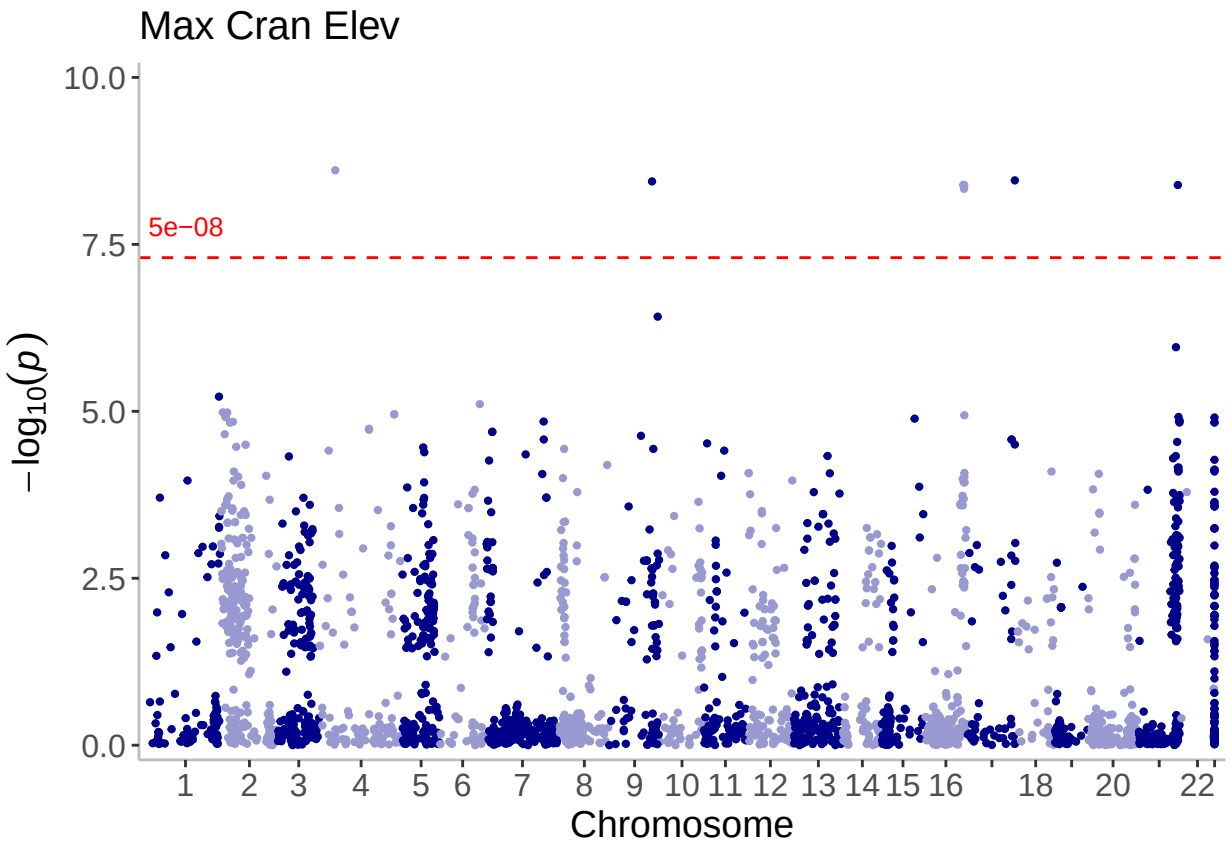
```
## [1] "There are no SNPs with p-values below 1e-05 in the input dataset. Use the [thresh] argument to
```



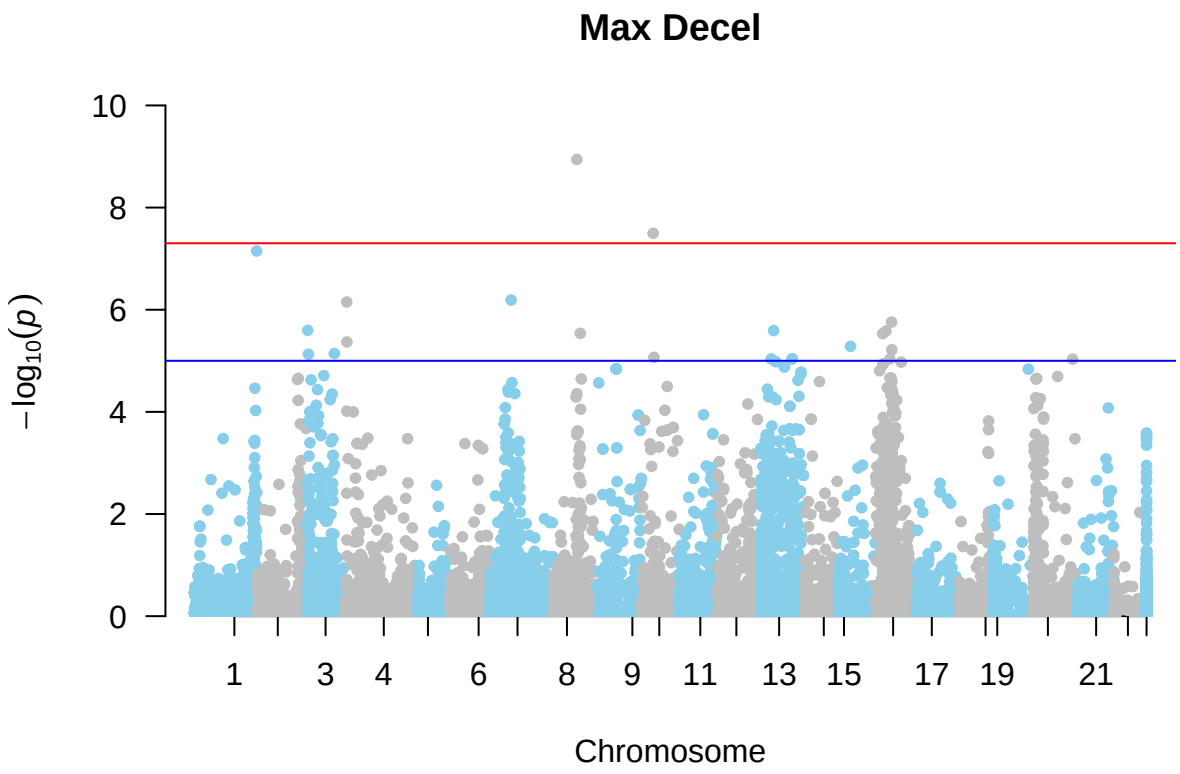

```
# maxCranElev - GWS!
qqman::manhattan(assoc.clean.maxCranElev, main = "Max Cran Elev", chr = "chrom",
    bp = "pos", p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y",
        "MT"), col = c("skyblue", "grey"))
```



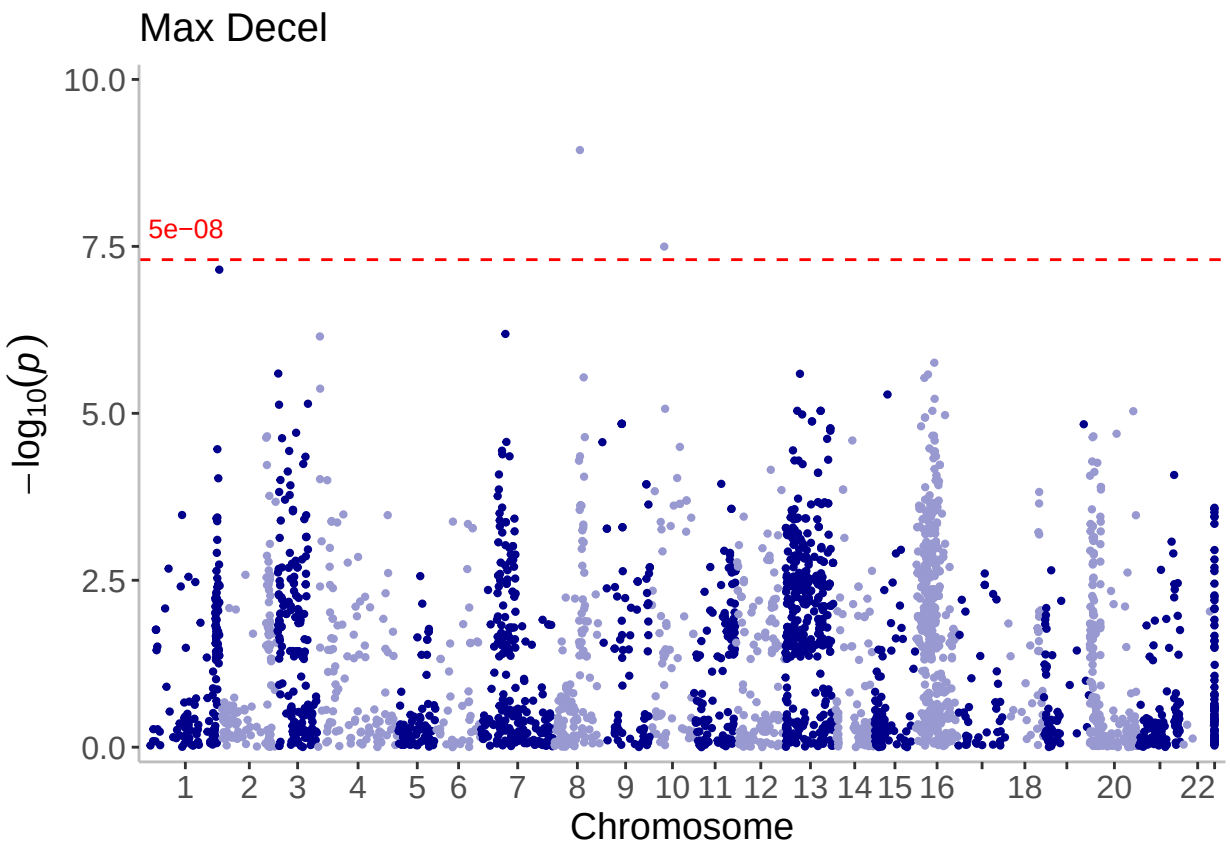

```
manhattan(assoc.clean.maxCranElev, title = "Max Cran Elev", ymin = 0, ymax = 10)
```



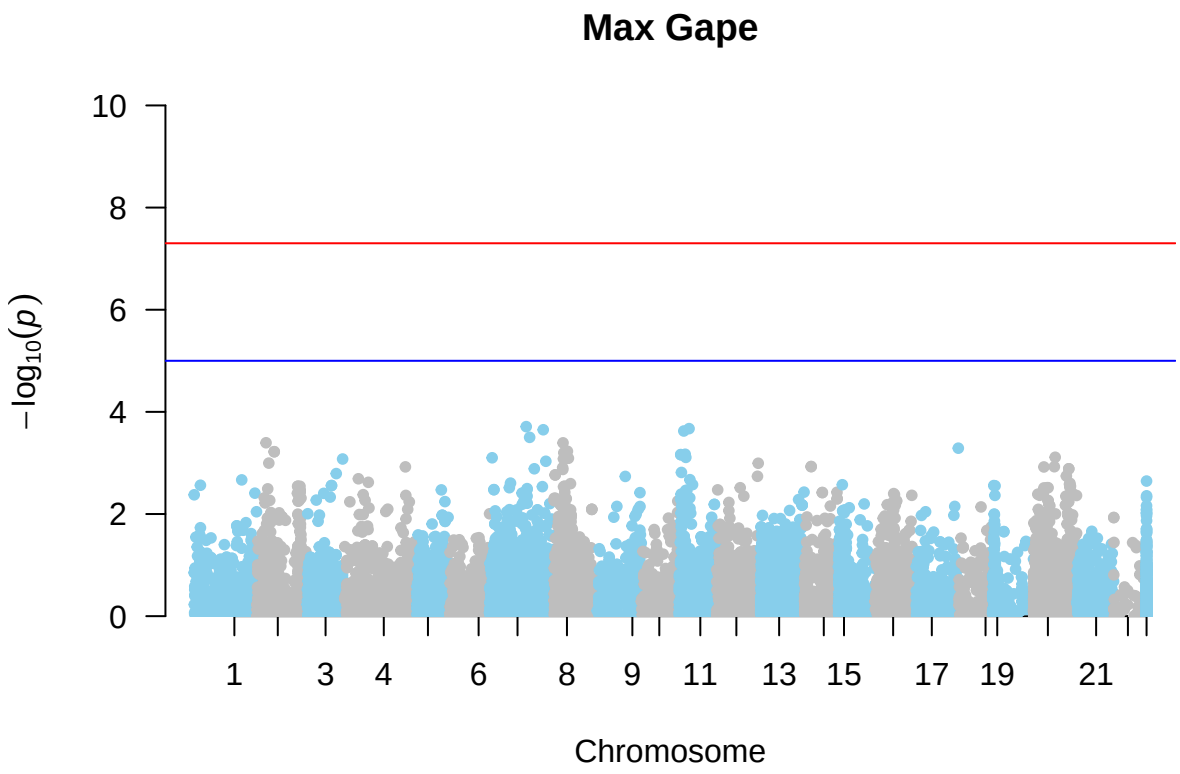

```
# maxDecel - GWS!
qqman::manhattan(assoc.clean.maxDecel, main = "Max Decel", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```



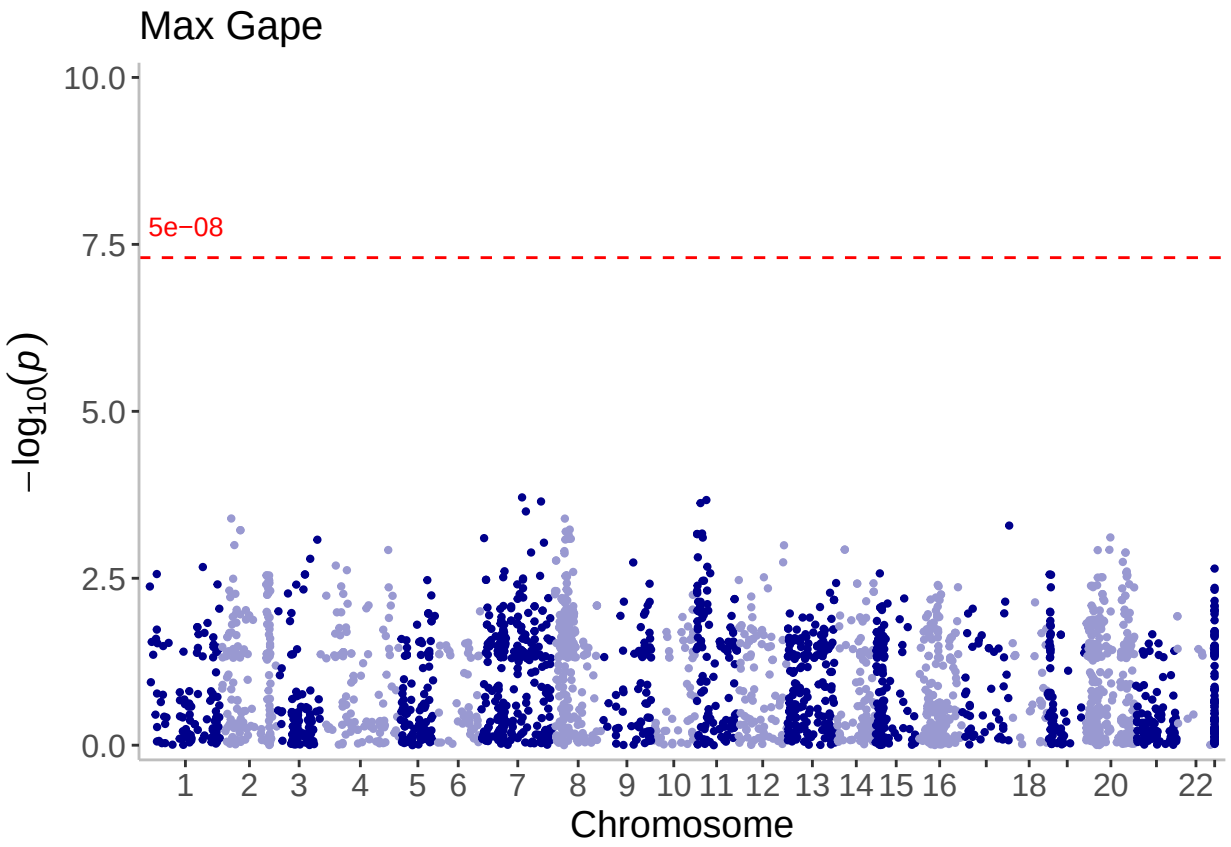

```
manhattan(assoc.clean.maxDecel, title = "Max Decel", ymin = 0, ymax = 10)
```



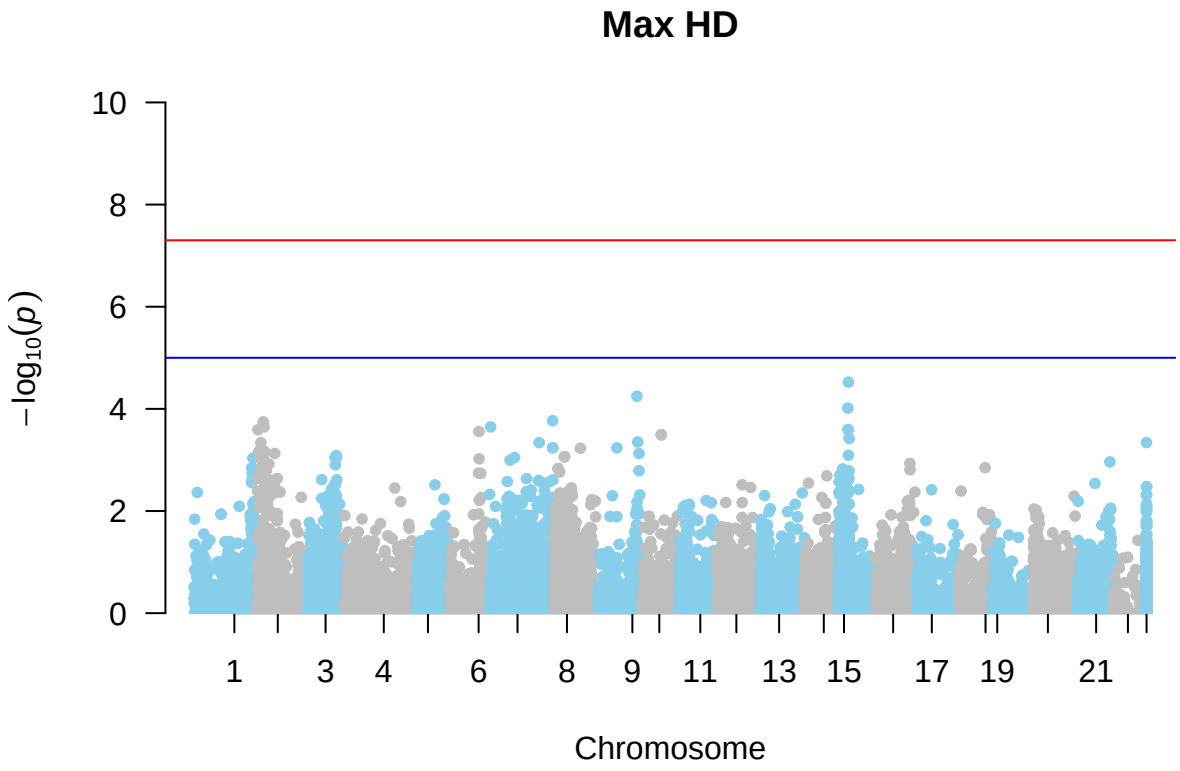

```
# maxGape - no GWS
qqman::manhattan(assoc.clean.maxGape, main = "Max Gape", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```



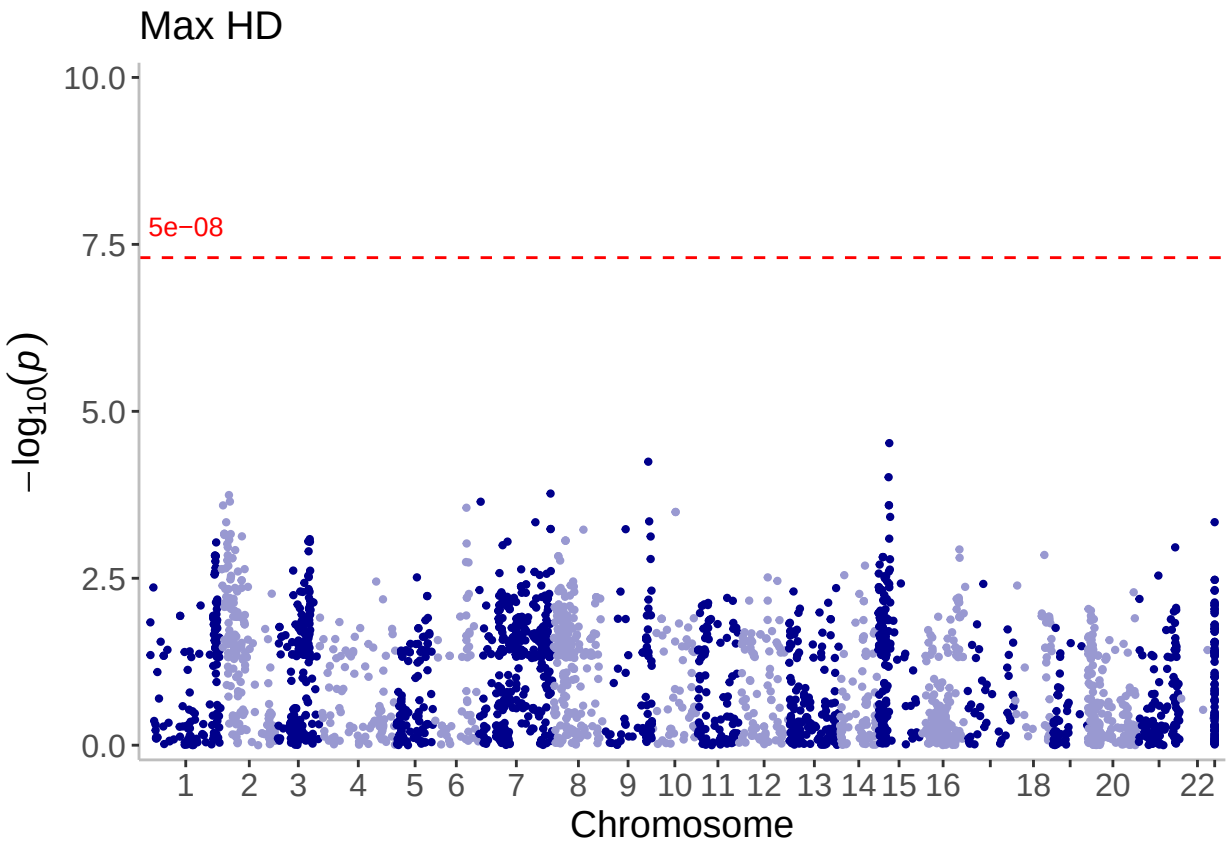

```
manhattan(assoc.clean.maxGape, title = "Max Gape", ymin = 0, ymax = 10)
```



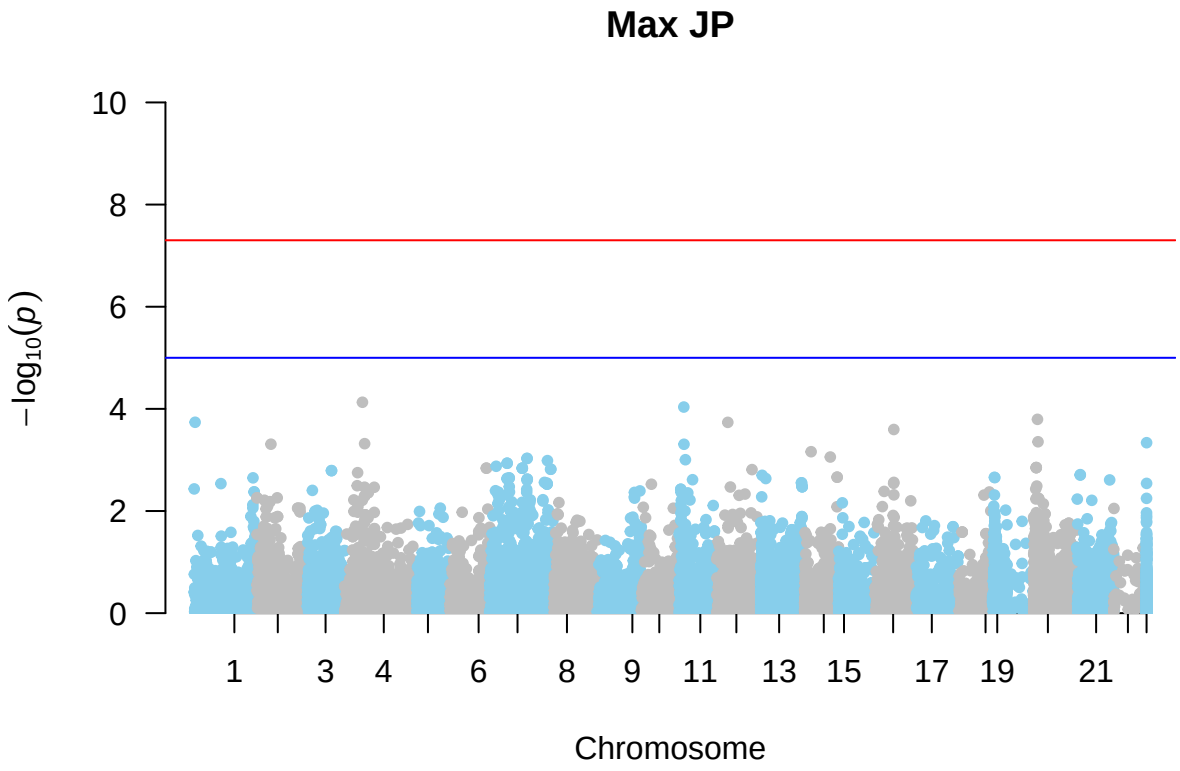

```
# maxHD - no GWS
qqman::manhattan(assoc.clean.maxHD, main = "Max HD", chr = "chrom", bp = "pos", p = "p",
    snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```



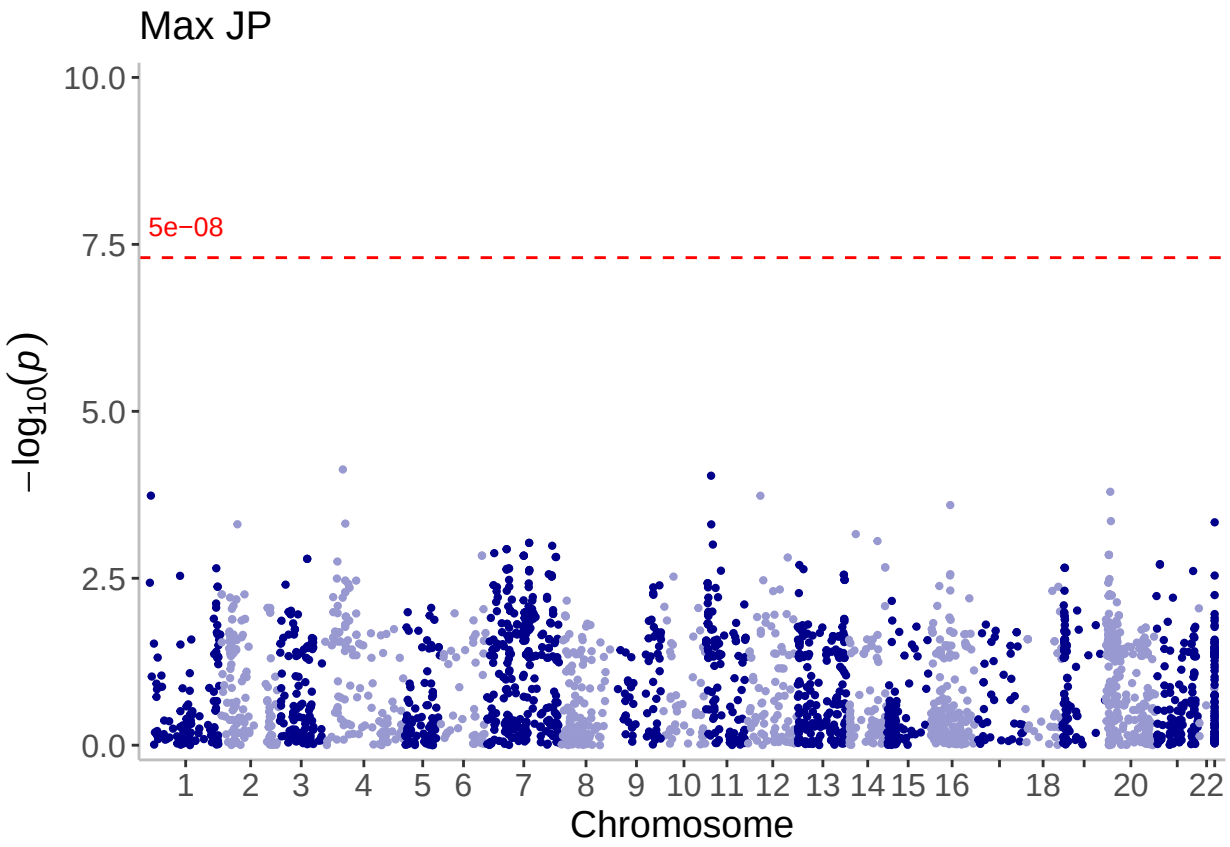

```
manhattan(assoc.clean.maxHD, title = "Max HD", ymin = 0, ymax = 10)
```



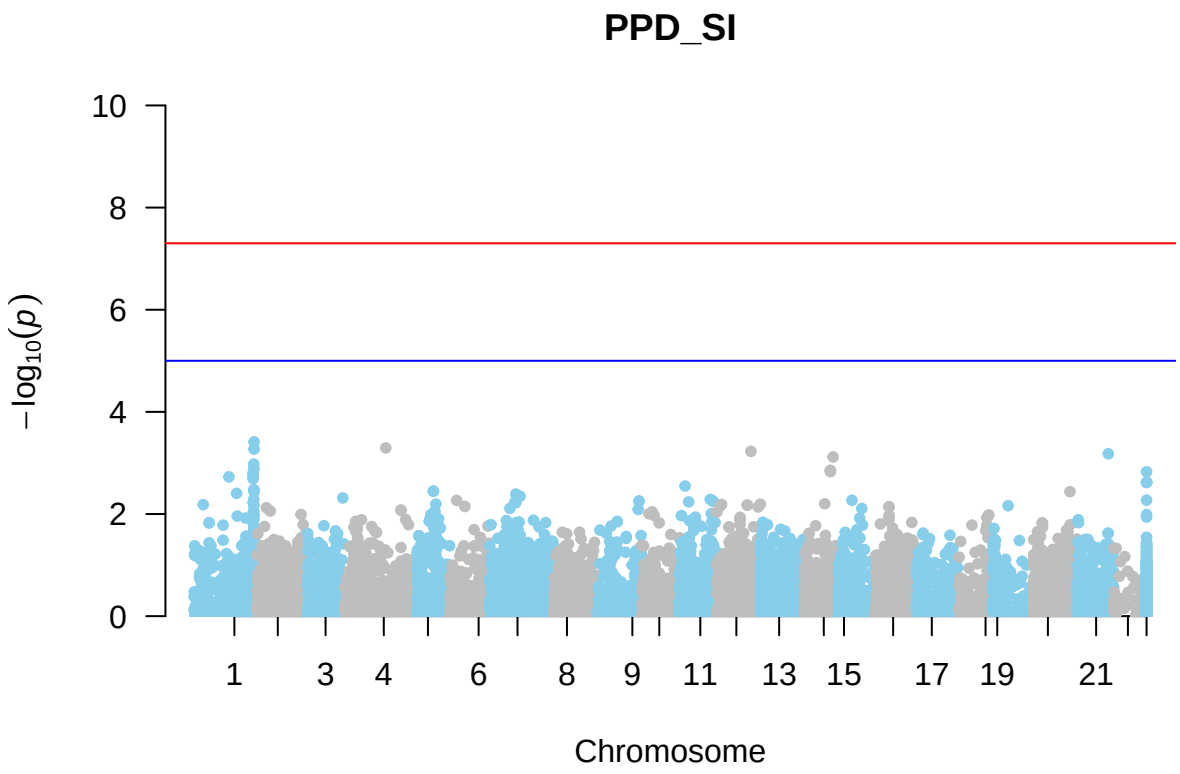

```
# maxJP - no GWS
qqman::manhattan(assoc.clean.maxJP, main = "Max JP", chr = "chrom", bp = "pos", p = "p",
    snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```



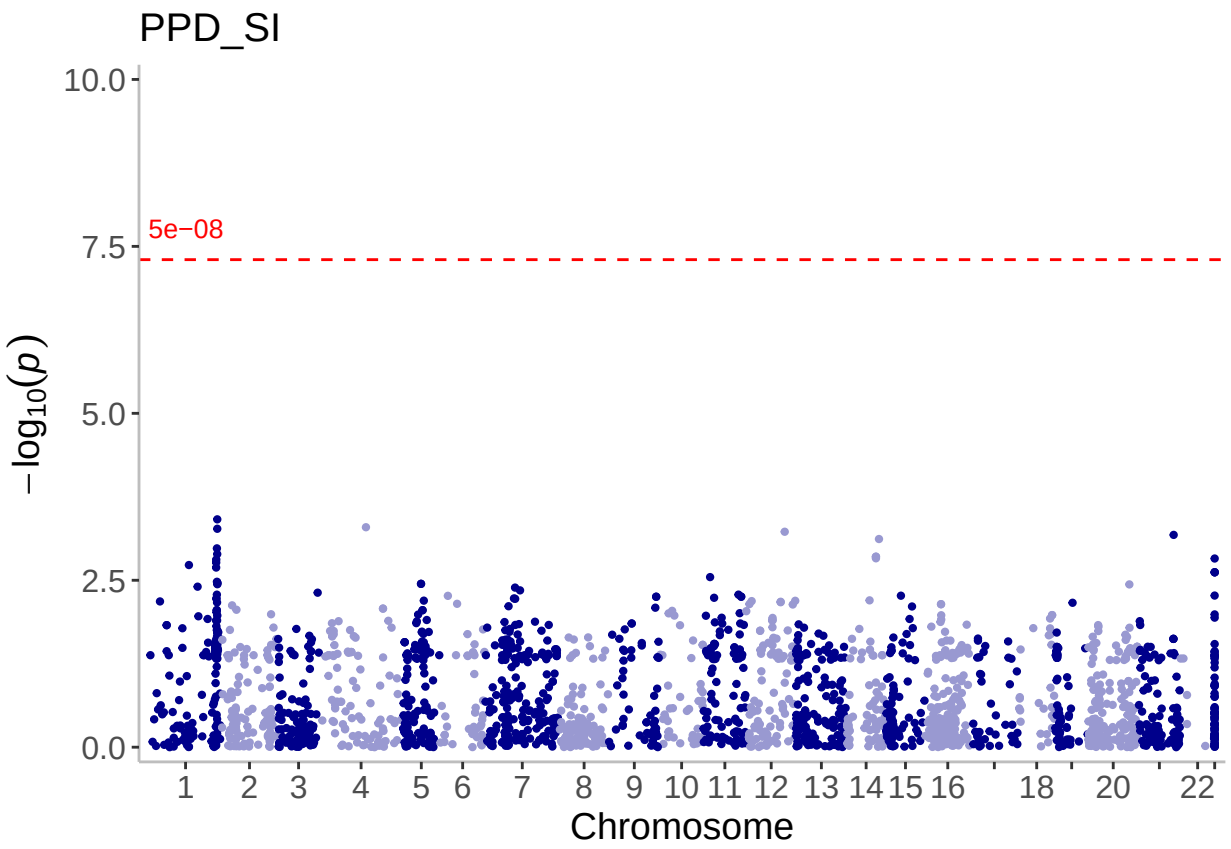

```
manhattan(assoc.clean.maxJP, title = "Max JP", ymin = 0, ymax = 10)
```



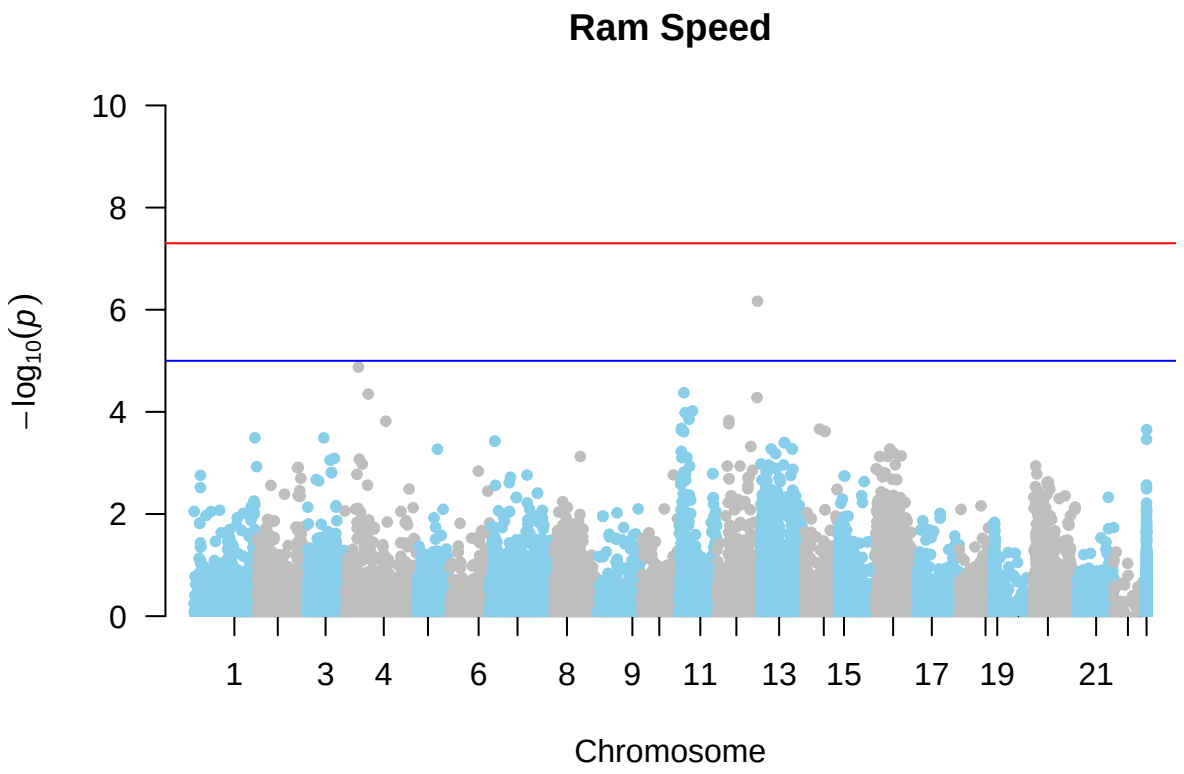

```
# PPD_SI - no GWS
qqman::manhattan(assoc.clean.PPD_SI, main = "PPD_SI", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```



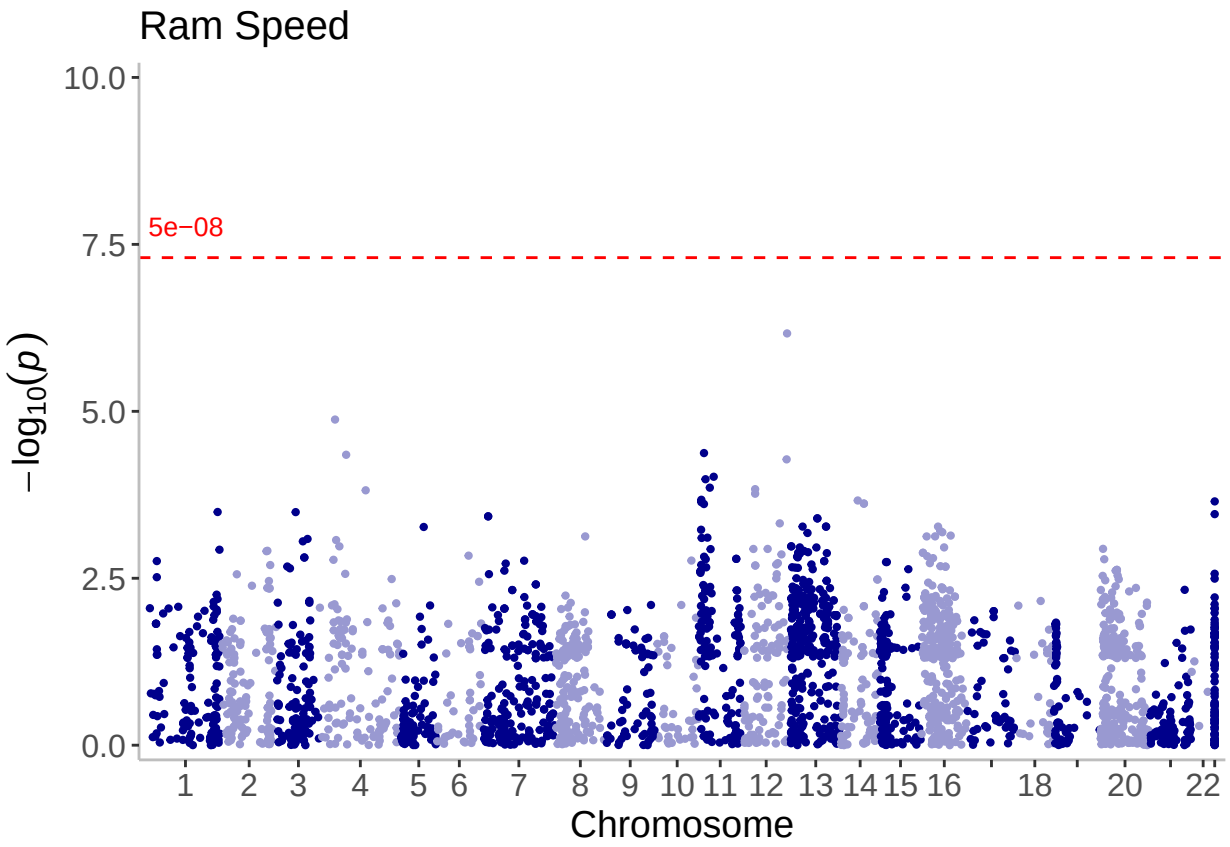

```
manhattan(assoc.clean.PPD_SI, title = "PPD_SI", ymin = 0, ymax = 10)
```



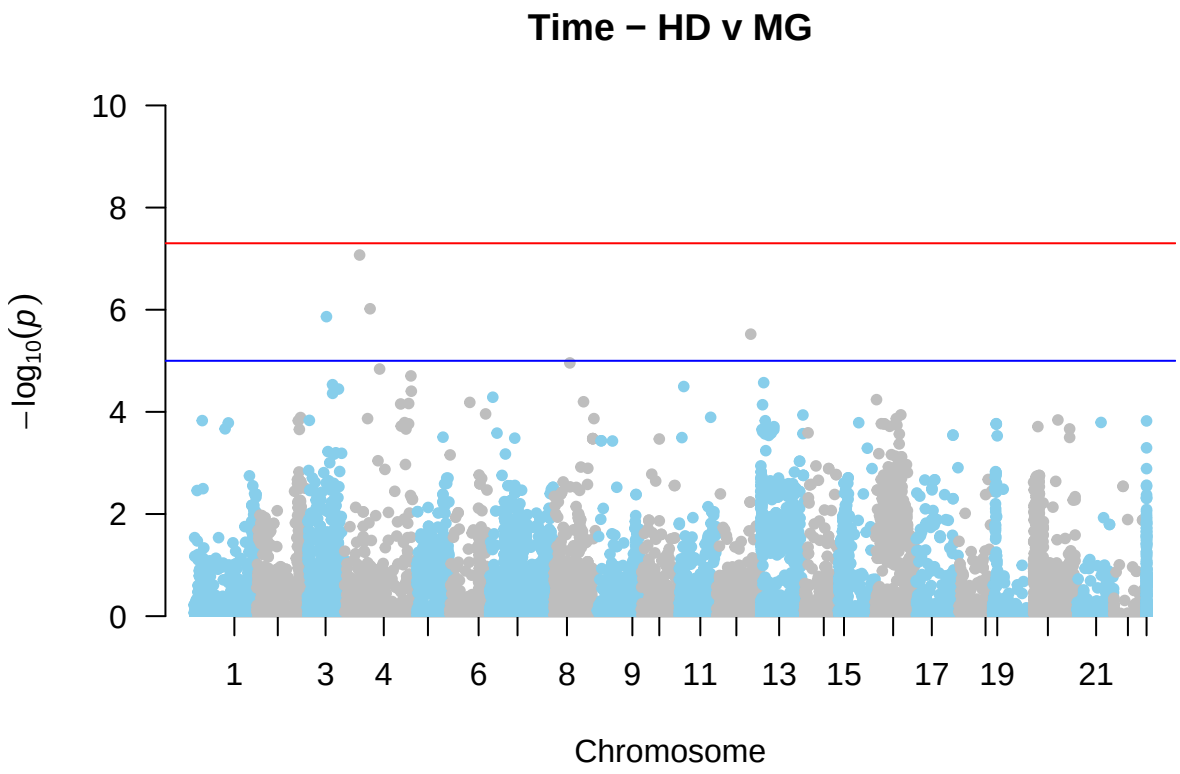

```
# ramSpeed - no GWS
qqman::manhattan(assoc.clean.ramSpeed, main = "Ram Speed", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```



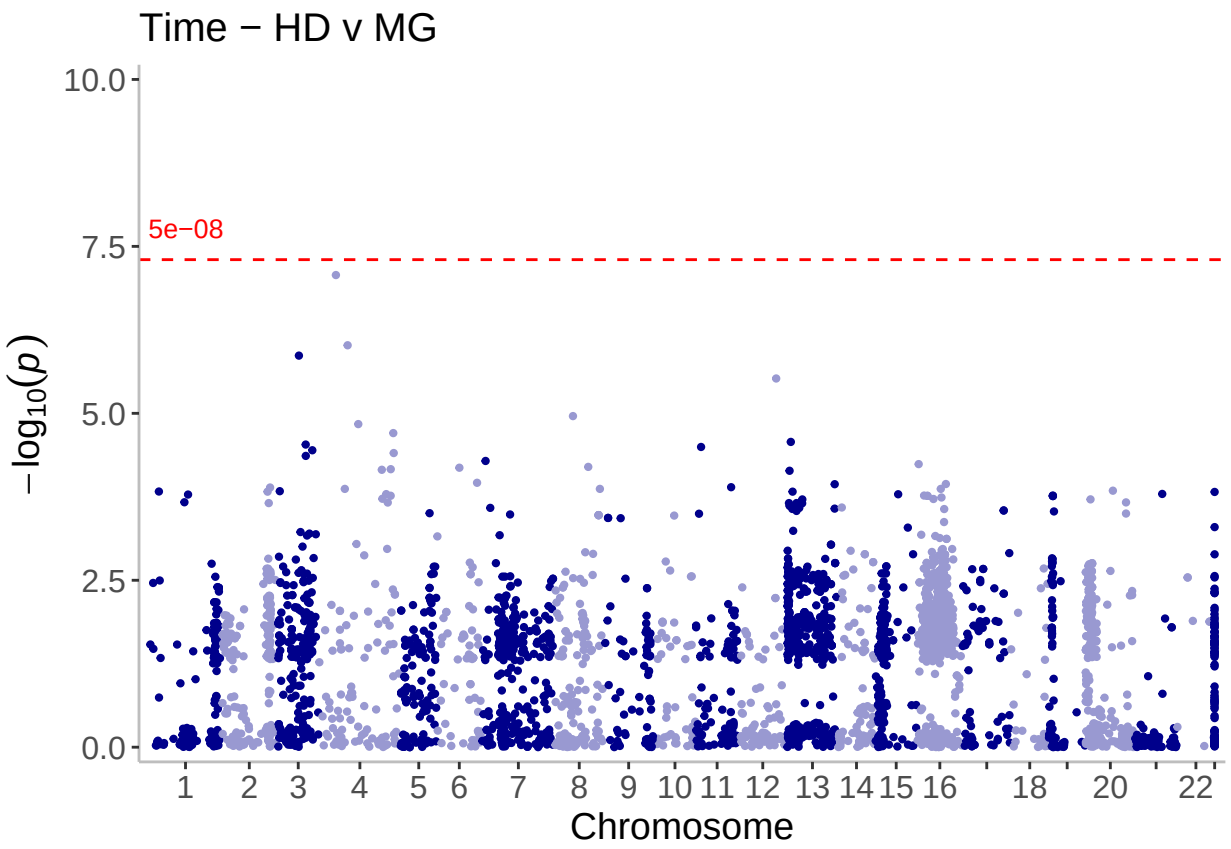

```
manhattan(assoc.clean.ramSpeed, title = "Ram Speed", ymin = 0, ymax = 10)
```



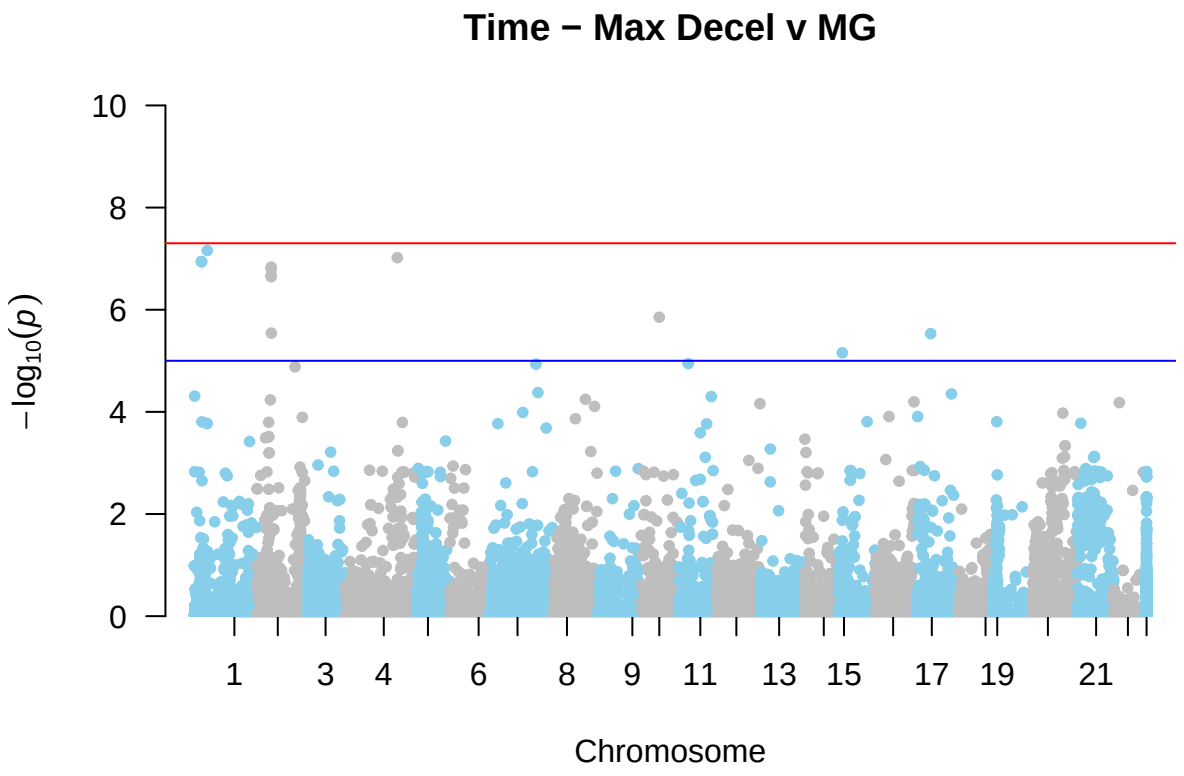

```
# time_HDvMG - very close to GWS, worth investigating
qqman::manhattan(assoc.clean.time_HDvMG, main = "Time - HD v MG", chr = "chrom",
    bp = "pos", p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y",
        "MT"), col = c("skyblue", "grey"))
```



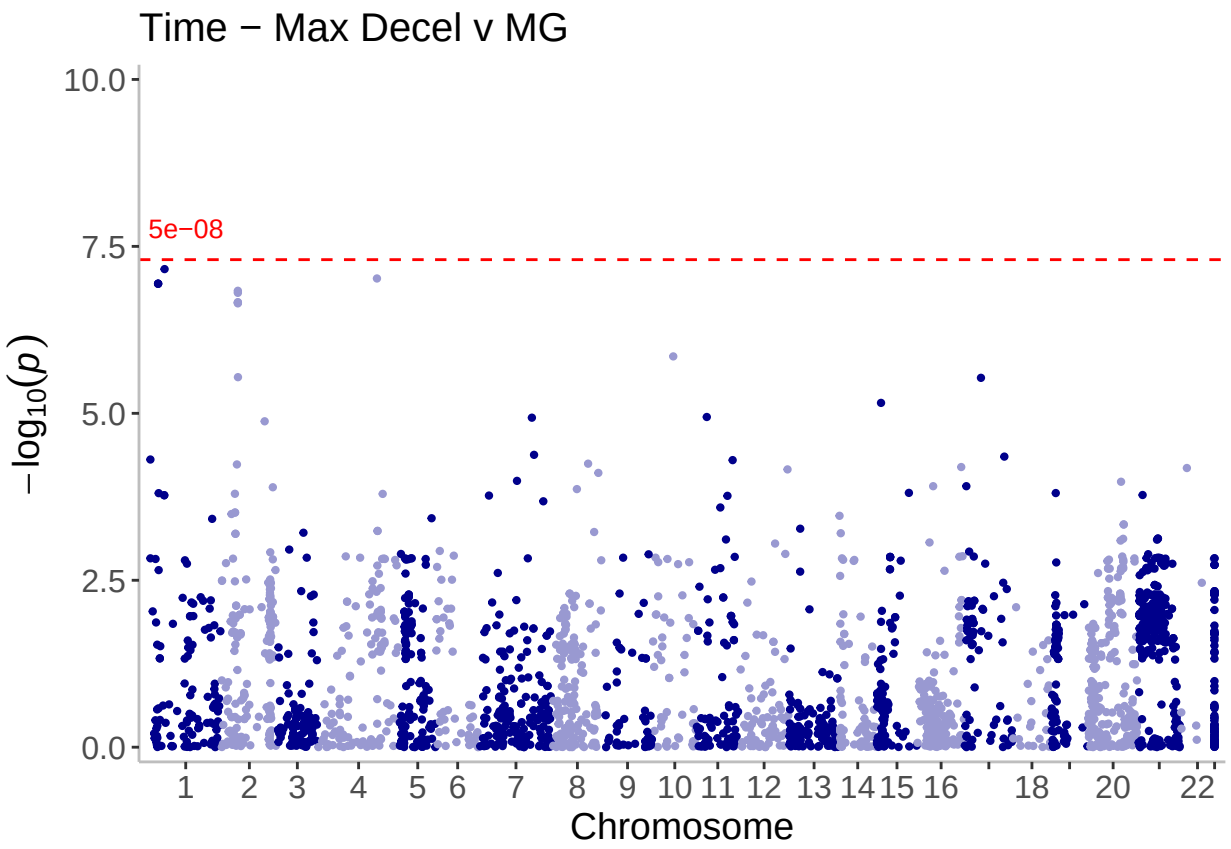

```
manhattan(assoc.clean.time_HDvMG, title = "Time - HD v MG", ymin = 0, ymax = 10)
```



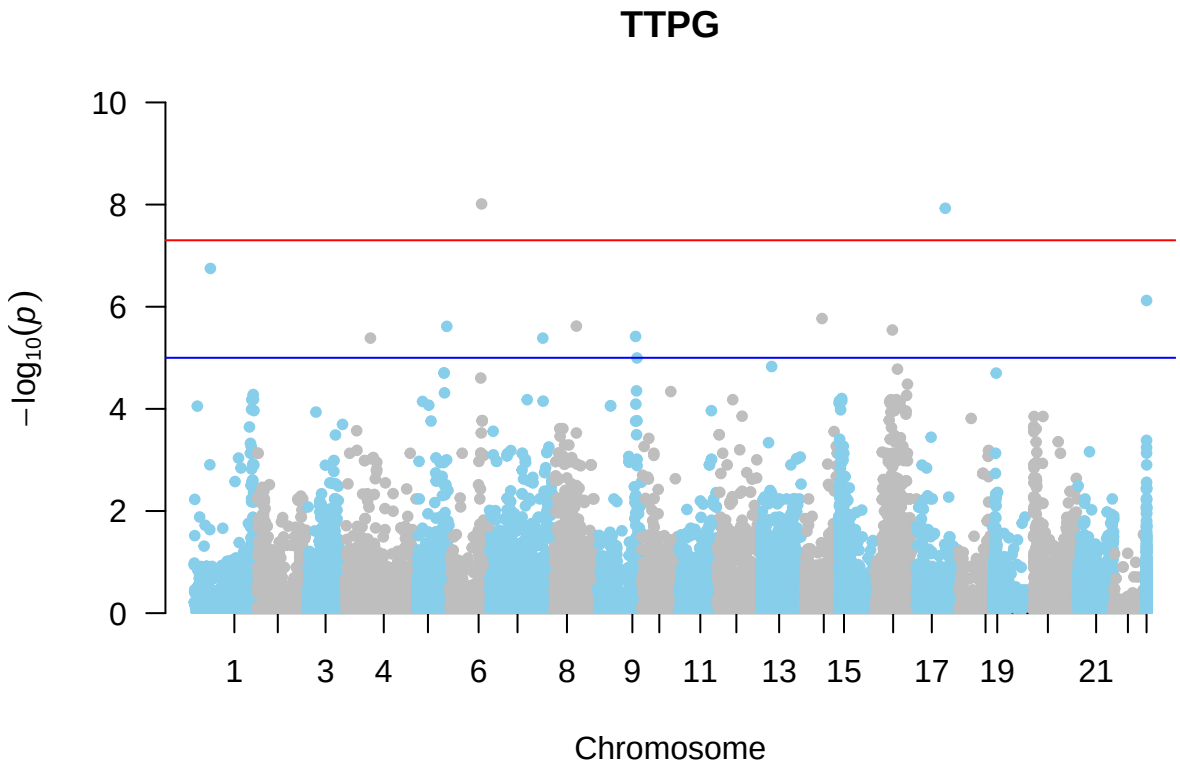

```
# time_maxDecelvMG - very close to GWS, worth investigating
qqman::manhattan(assoc.clean.time_maxDecelvMG, main = "Time - Max Decel v MG", chr = "chrom",
    bp = "pos", p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y",
        "MT"), col = c("skyblue", "grey"))
```



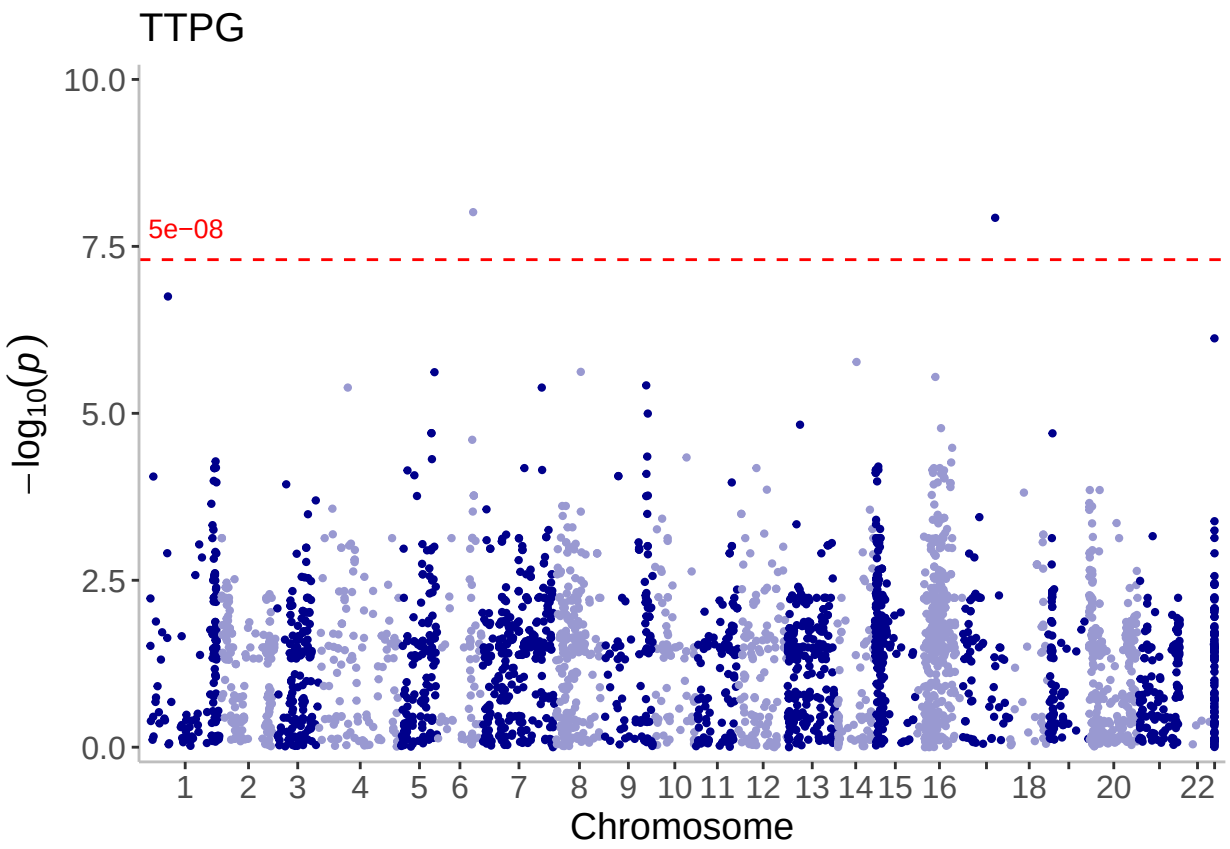

```
manhattan(assoc.clean.time_maxDecelvMG, title = "Time - Max Decel v MG", ymin = 0,
    ymax = 10)
```



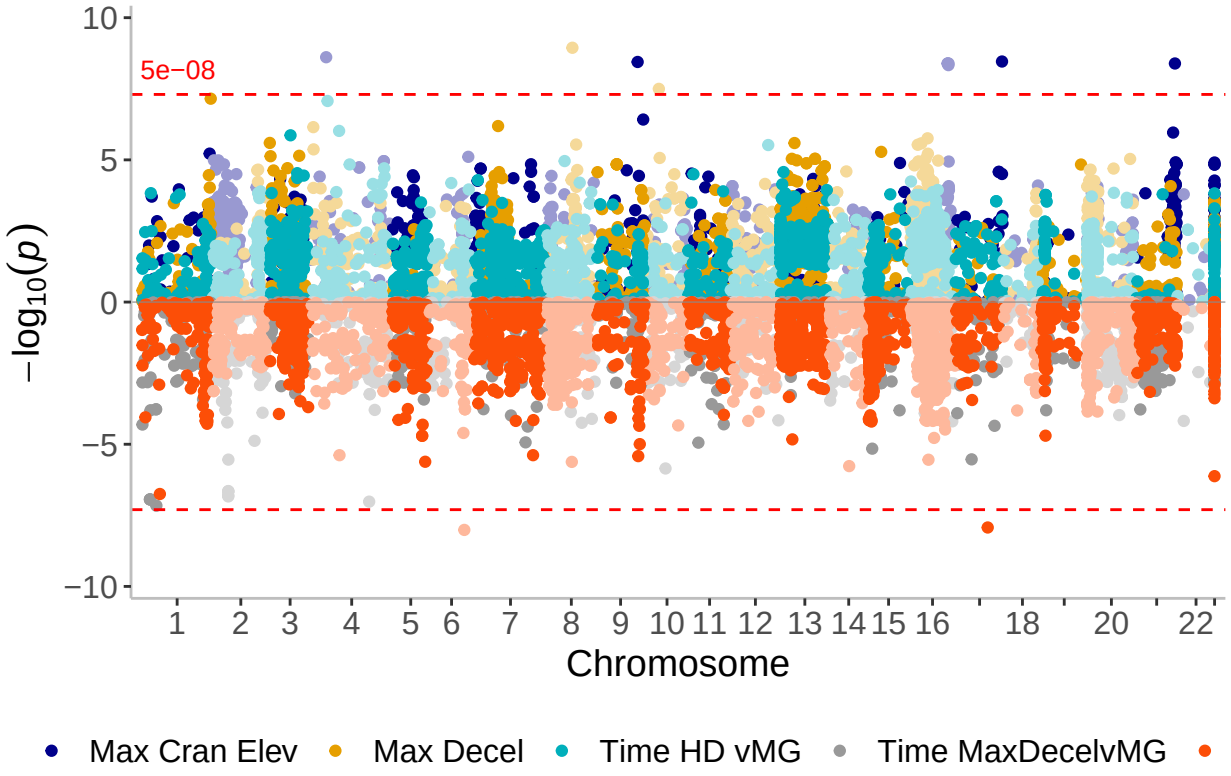

```
# ttpg - GWS!
qqman::manhattan(assoc.clean.ttpg, main = "TTPG", chr = "chrom", bp = "pos", p = "p",
    snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```



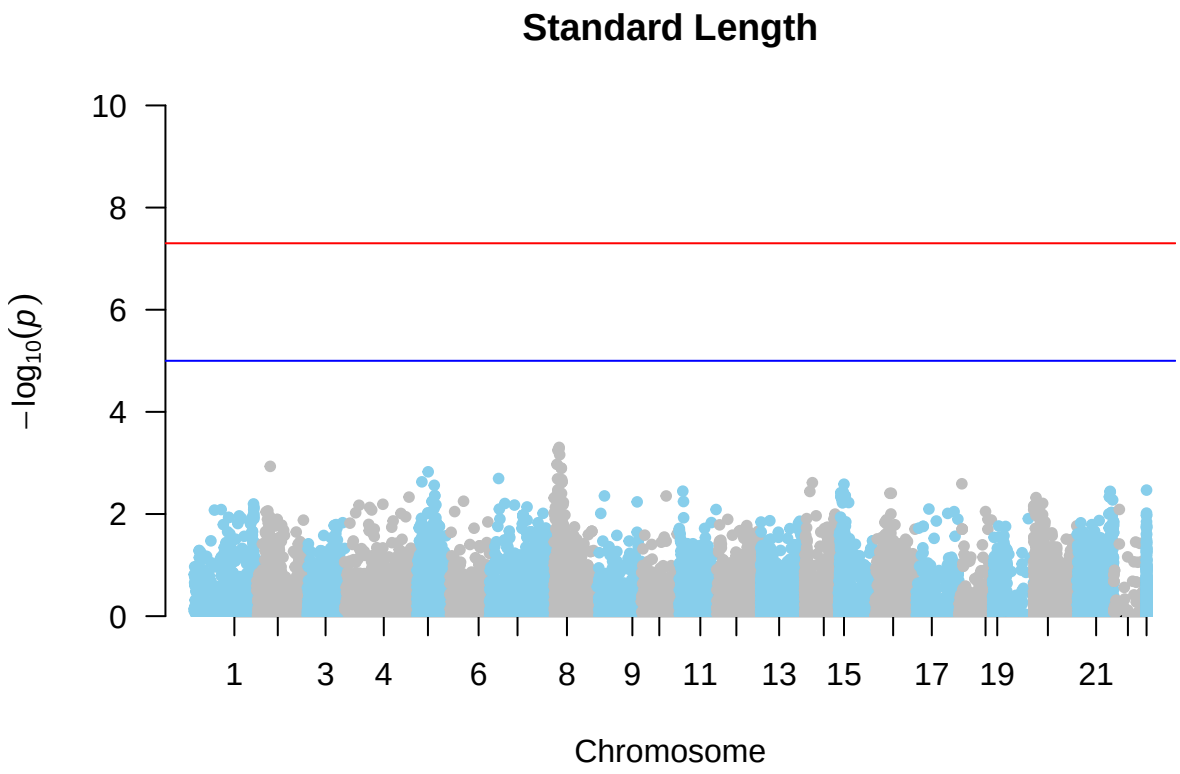

```
manhattan(assoc.clean.ttpg, title = "TTPG", ymin = 0, ymax = 10)
```




```
# sig or close to: maxCranElev, maxDecel, time_HDvMG, time_maxDecelvMG, ttpg
manhattan(list(assoc.clean.maxCranElev, assoc.clean.maxDecel, assoc.clean.time_HDvMG,
    assoc.clean.time_maxDecelvMG, assoc.clean.ttpg), ntop = 3, size = 1.5, legend_labels = c("Max Cran
    "Max Decel", "Time HD vMG", "Time MaxDecelvMG", "TTPG"), ymin = -10, ymax = 10)
```

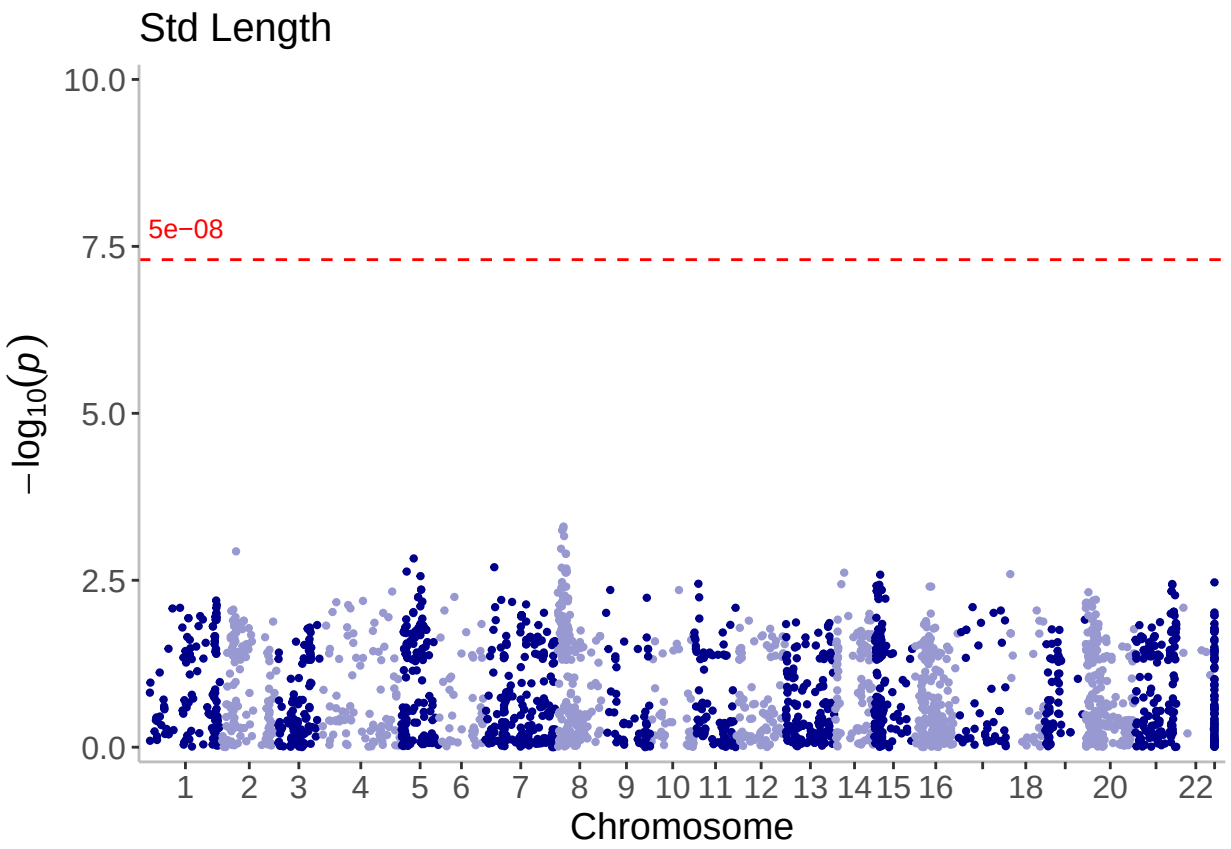



```
# length
qqman::manhattan(assoc.clean.length, main = "Standard Length", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```

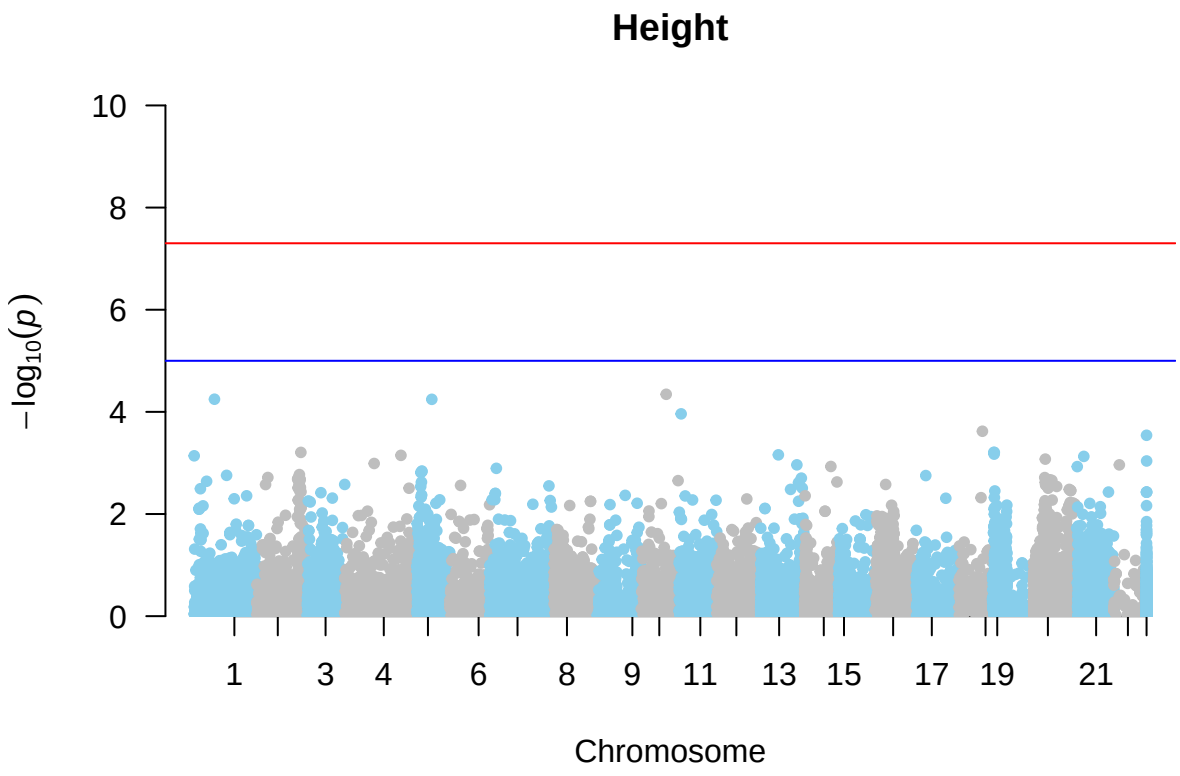



```
manhattan(assoc.clean.length, title = "Std Length", annotate = 1e-05, ymin = 0, ymax = 10)
```

```
## [1] "There are no SNPs with p-values below 1e-05 in the input dataset. Use the [thresh] argument to
```

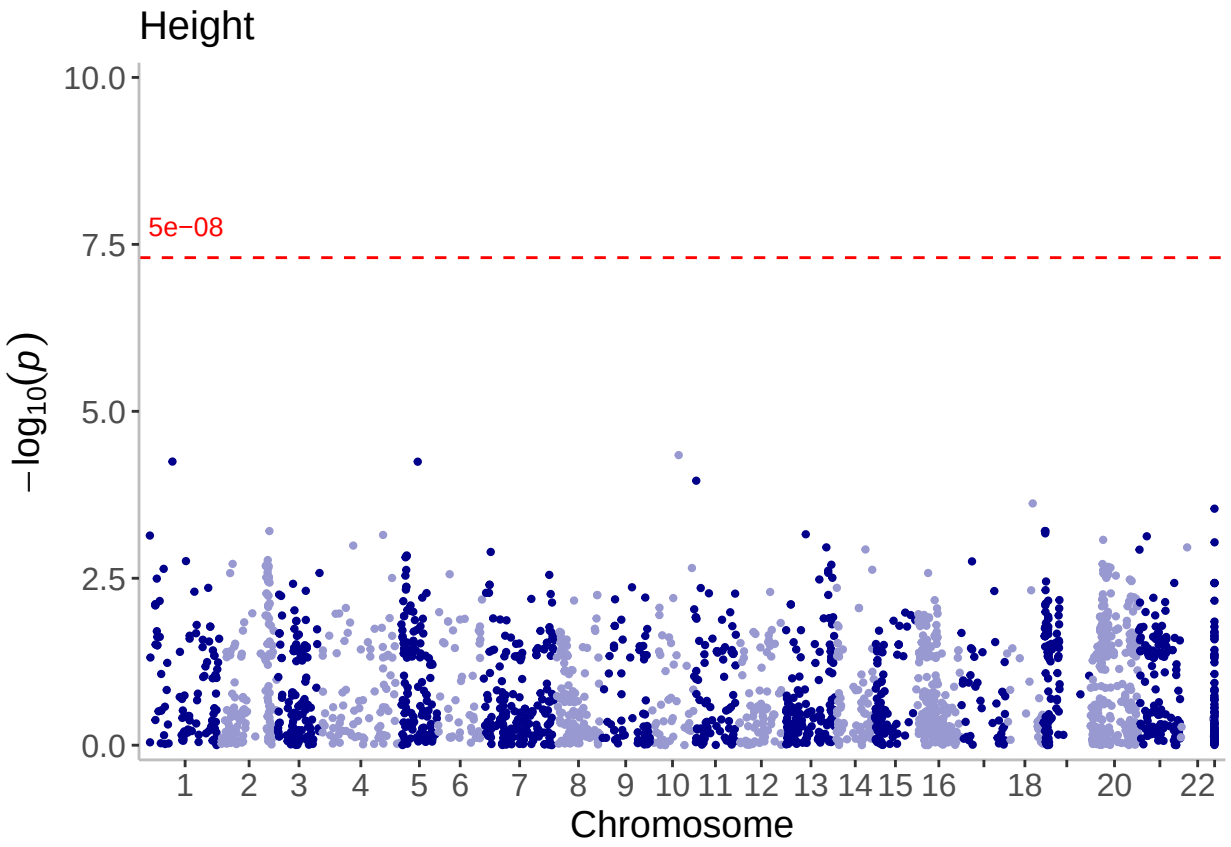



```
# height
qqman::manhattan(assoc.clean.height, main = "Height", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```

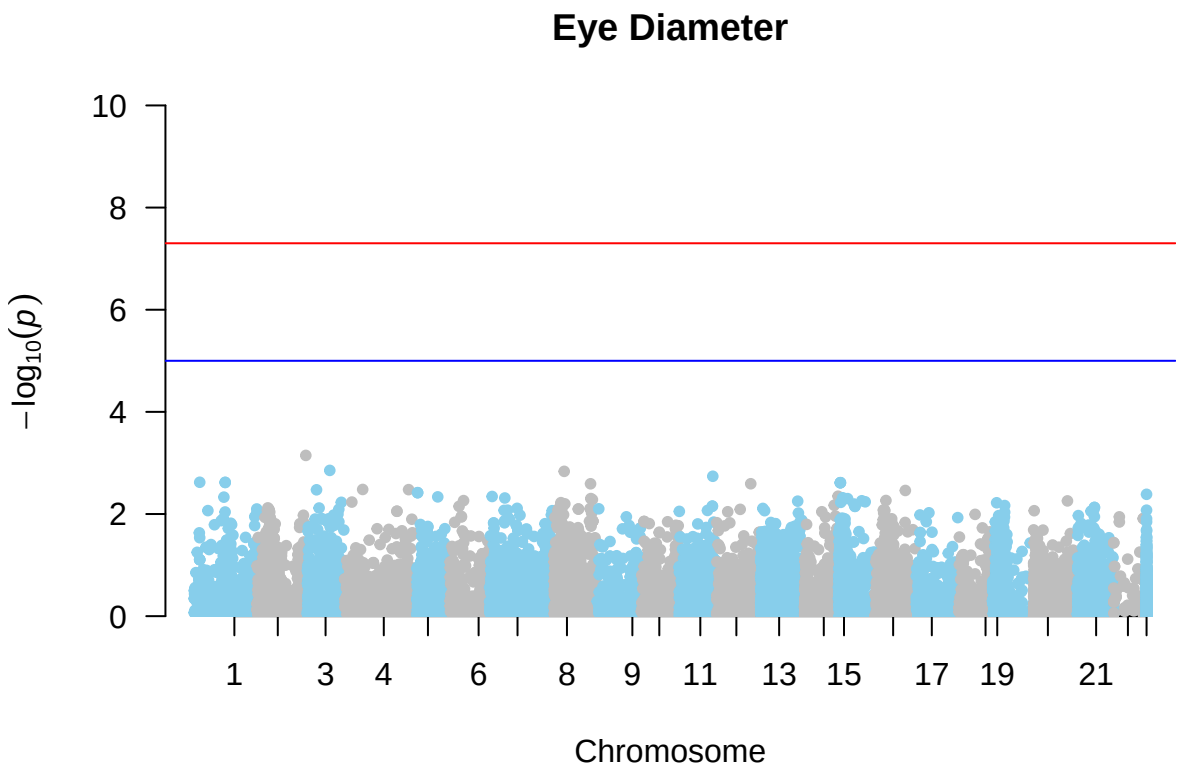



```
manhattan(assoc.clean.height, title = "Height", annotate = 1e-05, ymin = 0, ymax = 10)
```

```
## [1] "There are no SNPs with p-values below 1e-05 in the input dataset. Use the [thresh] argument to
```

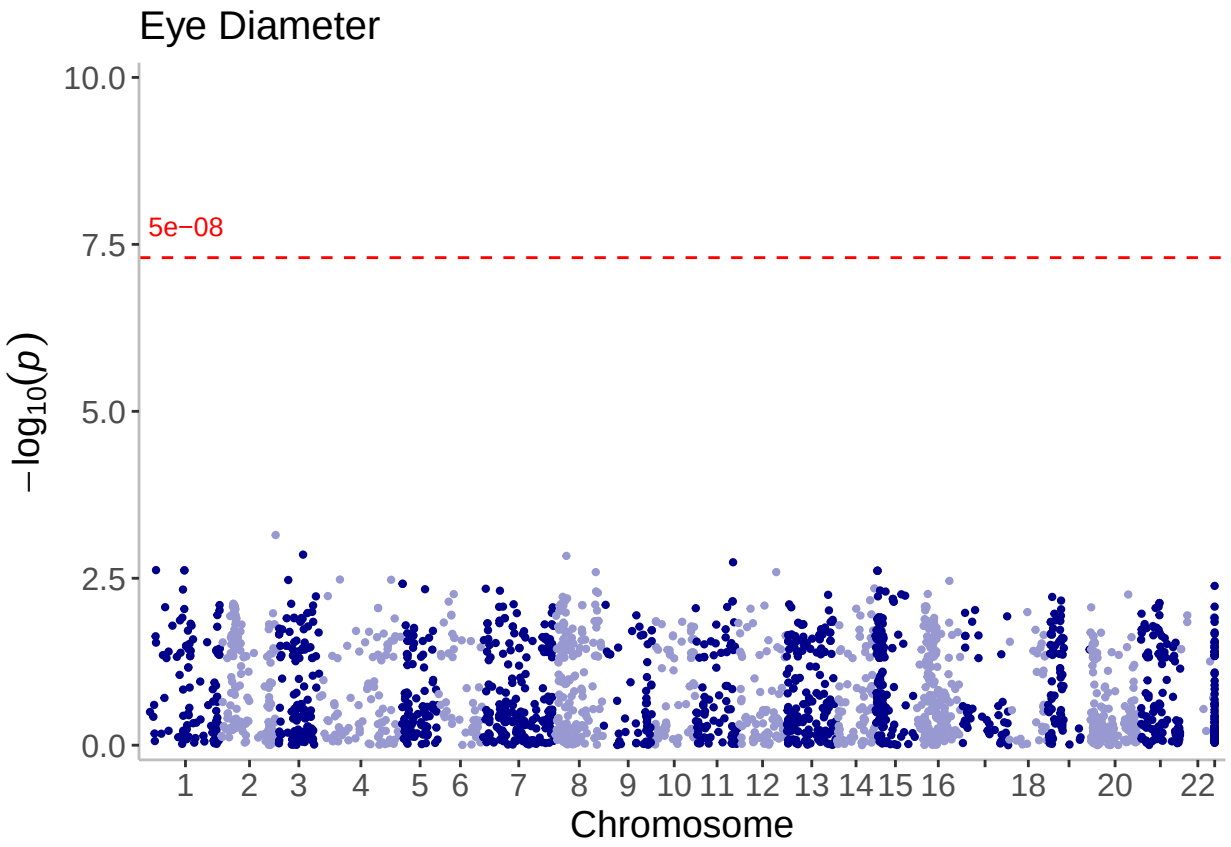



```
# eye diameter
qqman::manhattan(assoc.clean.eye.dia, main = "Eye Diameter", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```

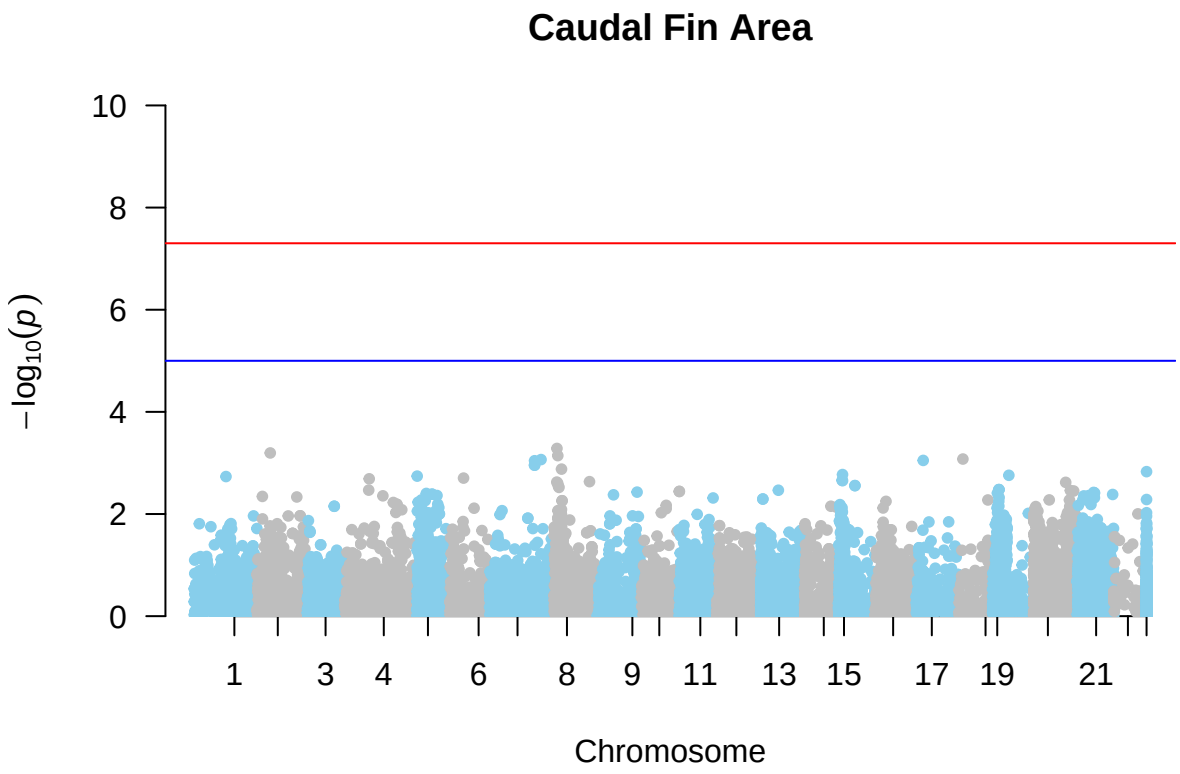



```
manhattan(assoc.clean.eye.dia, title = "Eye Diameter", annotate = 1e-05, ymin = 0,
    ymax = 10)
```

```
## [1] "There are no SNPs with p-values below 1e-05 in the input dataset. Use the [thresh] argument to
```

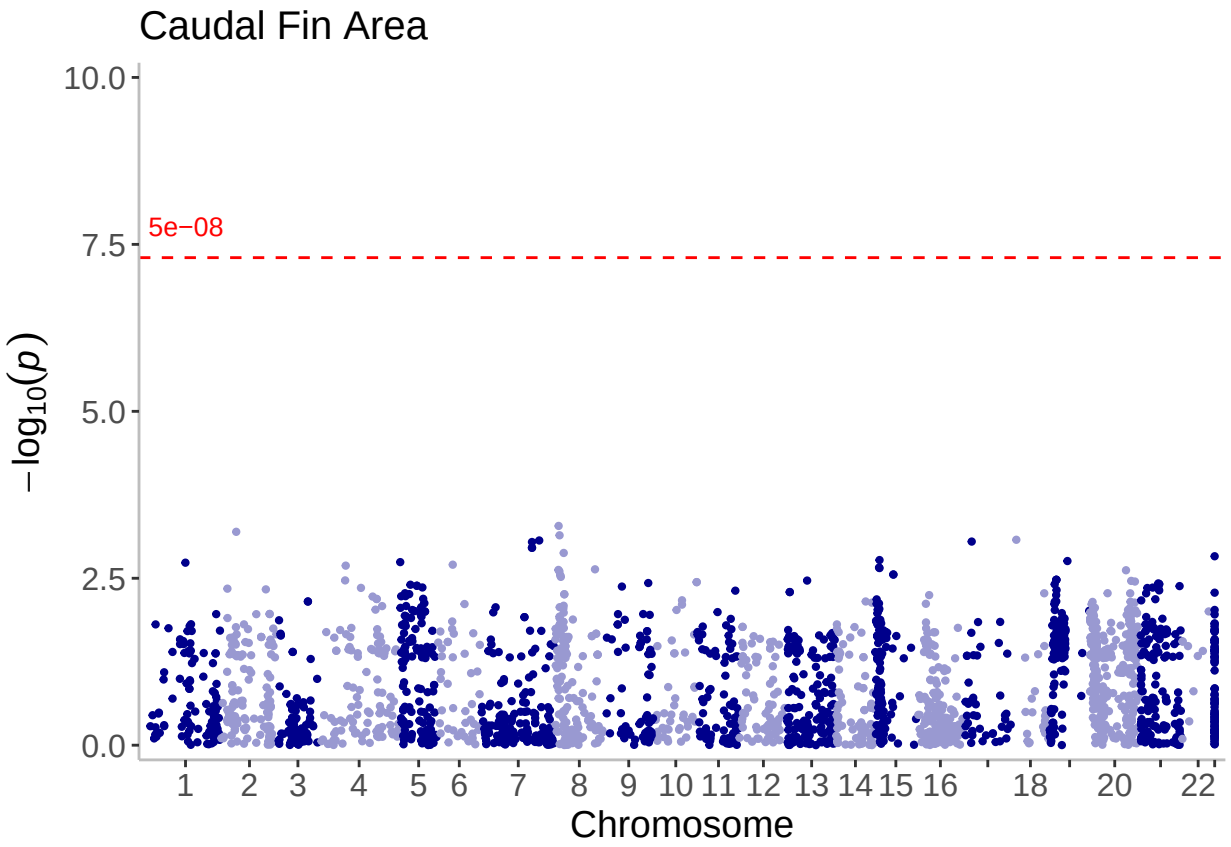



```
# caudal fin area
qqman::manhattan(assoc.clean.caudal.area, main = "Caudal Fin Area", chr = "chrom",
    bp = "pos", p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y",
        "MT"), col = c("skyblue", "grey"))
```

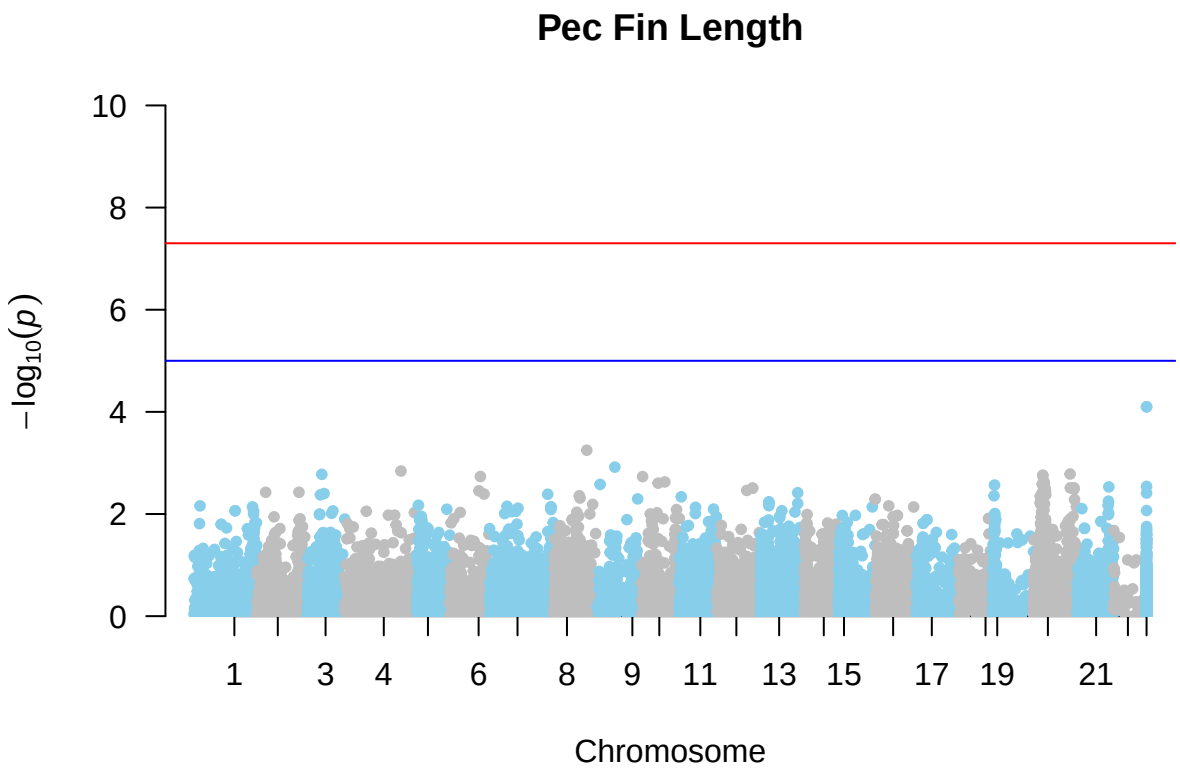



```
manhattan(assoc.clean.caudal.area, title = "Caudal Fin Area", annotate = 1e-05, ymin = 0,
    ymax = 10)
```

```
## [1] "There are no SNPs with p-values below 1e-05 in the input dataset. Use the [thresh] argument to l
```

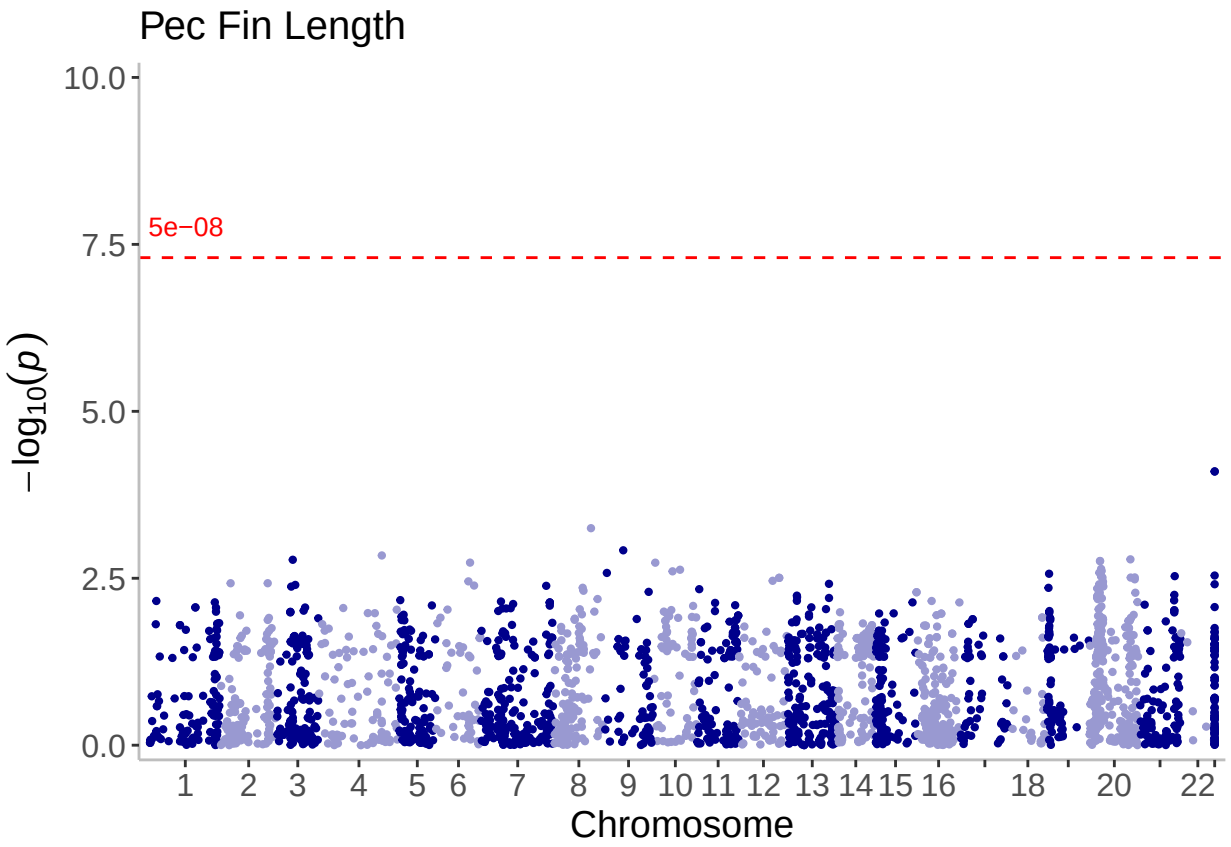



```
# pec fin length
qqman::manhattan(assoc.clean.pec.length, main = "Pec Fin Length", chr = "chrom",
    bp = "pos", p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y",
        "MT"), col = c("skyblue", "grey"))
```

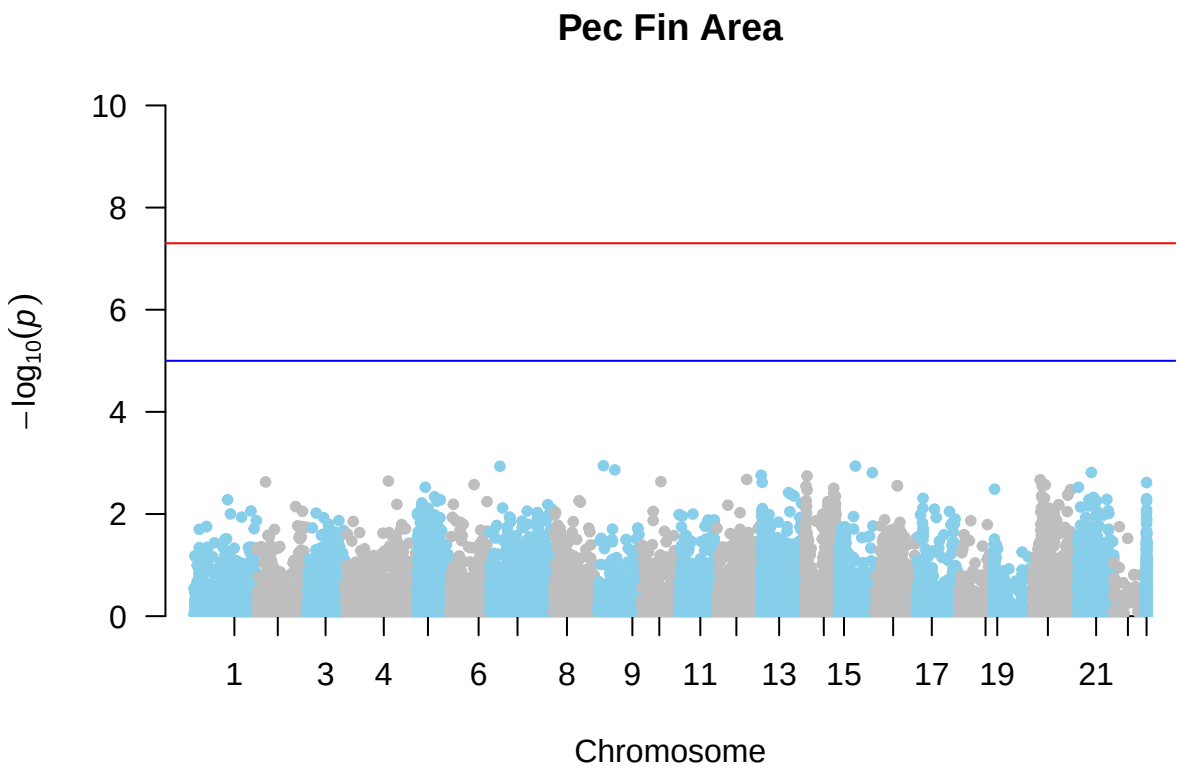



```
manhattan(assoc.clean.pec.length, title = "Pec Fin Length", annotate = 1e-05, ymin = 0,
    ymax = 10)
```

```
## [1] "There are no SNPs with p-values below 1e-05 in the input dataset. Use the [thresh] argument to
```

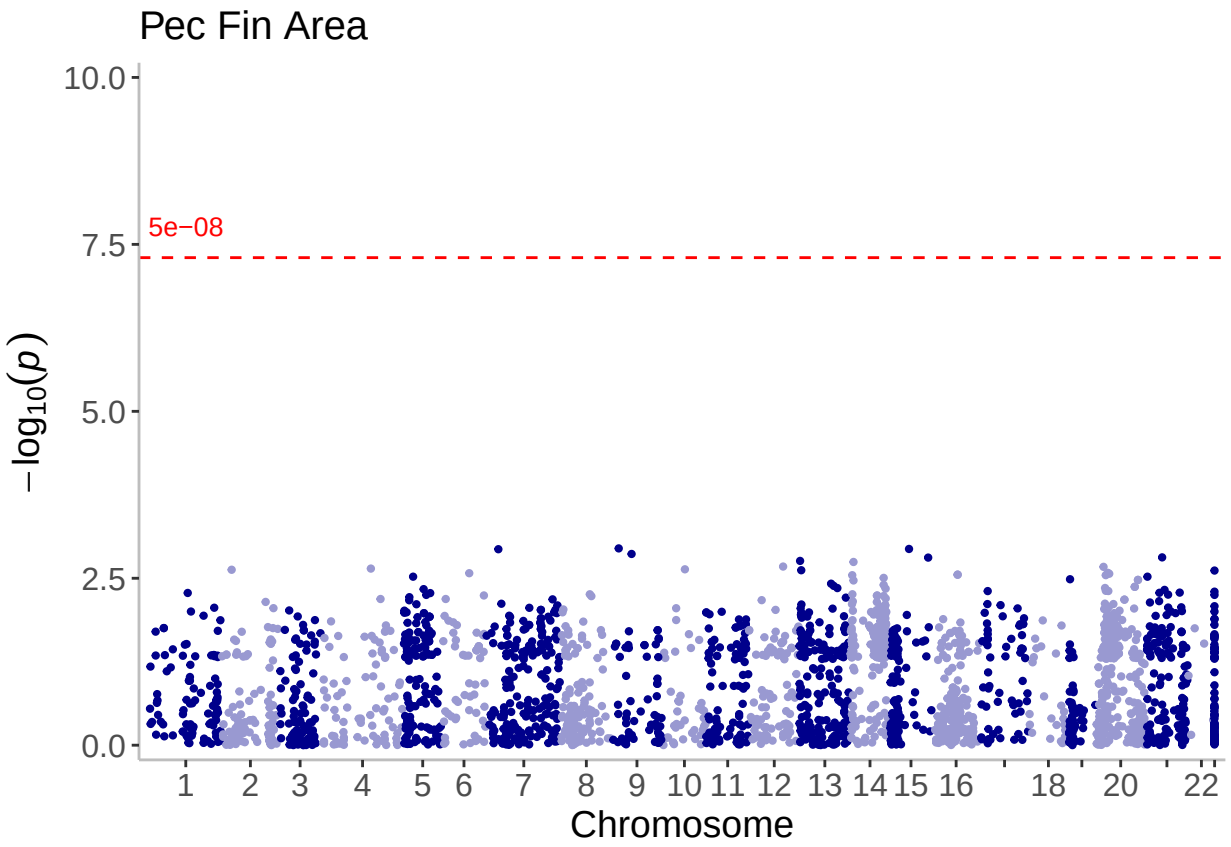



```
# pec fin area
qqman::manhattan(assoc.clean.pec.area, main = "Pec Fin Area", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```

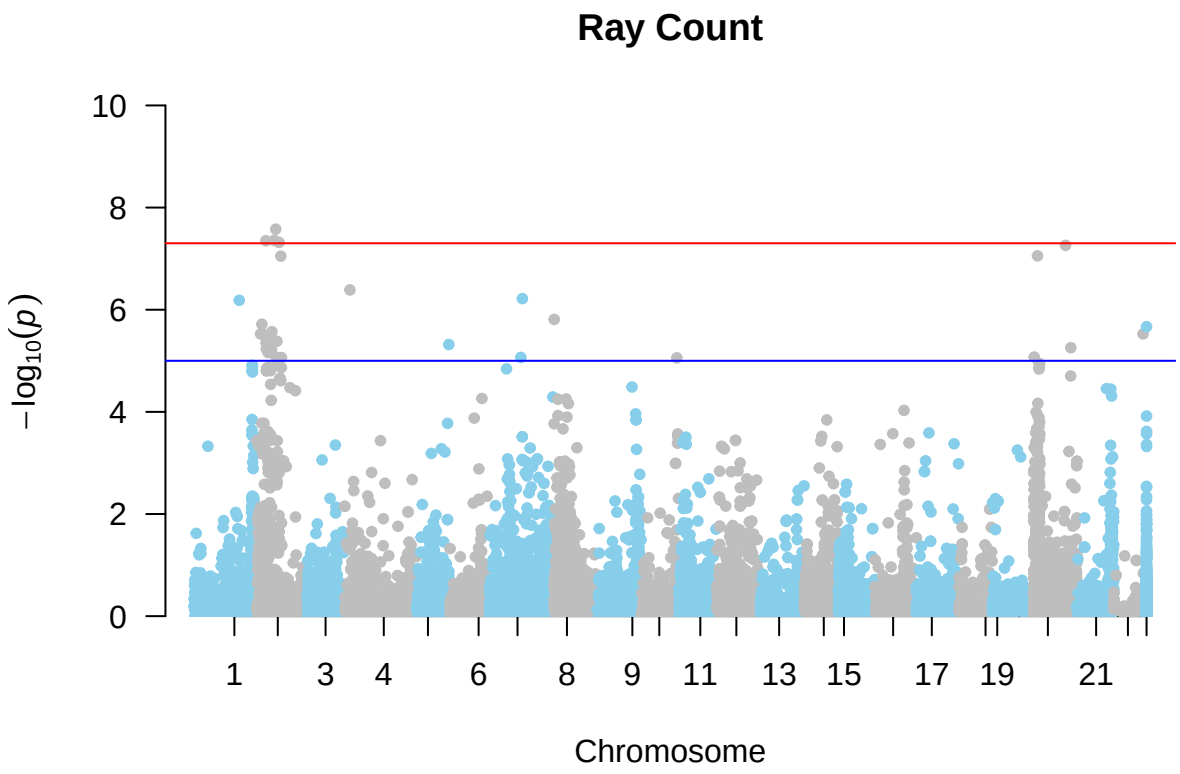



```
manhattan(assoc.clean.pec.area, title = "Pec Fin Area", annotate = 1e-05, ymin = 0,
    ymax = 10)
```

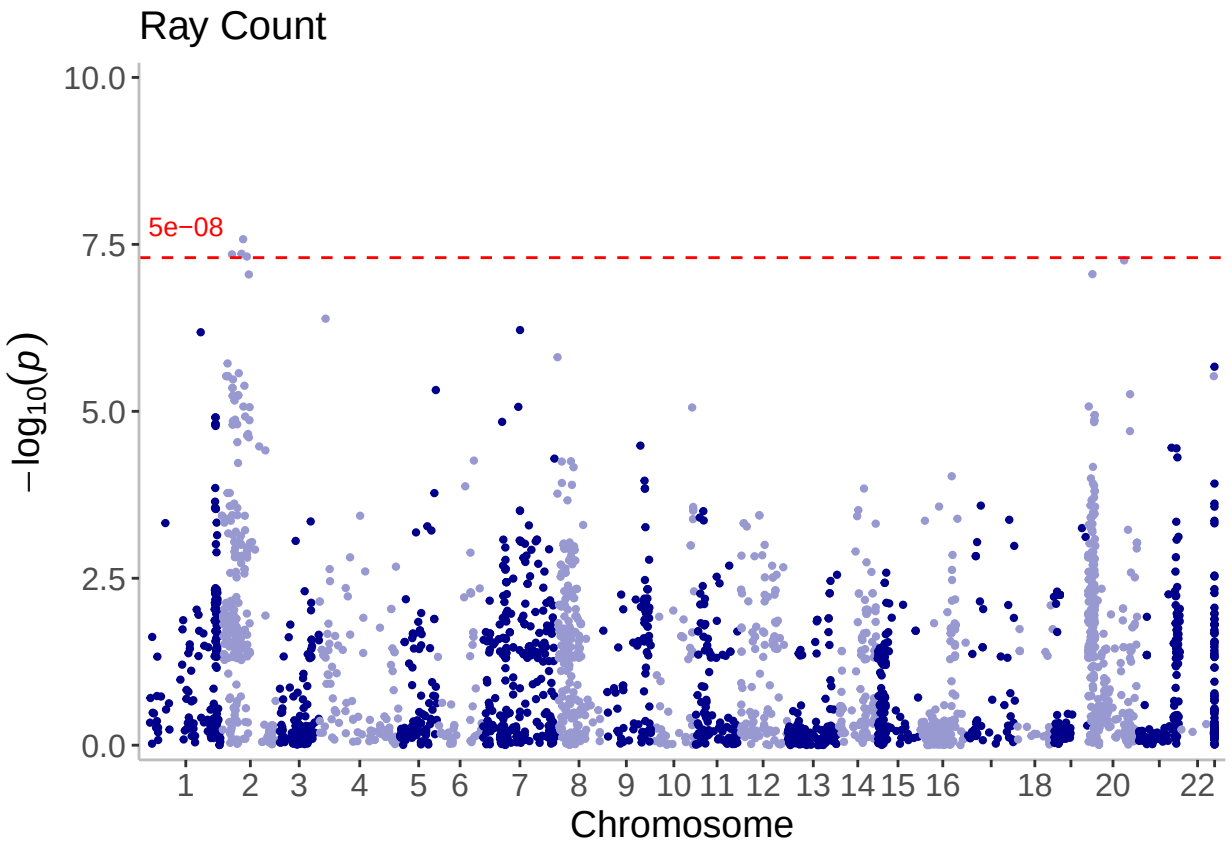
```
## [1] "There are no SNPs with p-values below 1e-05 in the input dataset. Use the [thresh] argument to
```




```
# ray count - only morph pheno with GWS
qqman::manhattan(assoc.clean.ray, main = "Ray Count", chr = "chrom", bp = "pos",
    p = "p", snp = "variant.id", ylim = c(0, 10), chrlabs = c(1:21, "Y", "MT"), col = c("skyblue",
        "grey"))
```




```
manhattan(assoc.clean.ray, title = "Ray Count", ymin = 0, ymax = 10)
```



Significant SNPs

```
threshold <- 5e-08

mce.sig <- assoc.clean.maxCranElev %>%
    filter(p < threshold)

md.sig <- assoc.clean.maxDecel %>%
    filter(p < threshold)

time.hdvmg.sugg.sig <- assoc.clean.time_HDvMG %>%
    filter(p < 1e-05)

time.mdvmg.sugg.sig <- assoc.clean.time_maxDecelvMG %>%
    filter(p < 1e-05)

ttpg.sig <- assoc.clean.ttpg %>%
    filter(p < threshold)

ray.sig <- assoc.clean.ray %>%
    filter(p < threshold)
```

(suggestive for the time variables)

Linkage Disequilibrium Analysis in SNPs with genome-wide significance

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

```
files.pvar <- c("maxCranElev", "maxDecel", "time_HDvMG", "time_maxDecelvMG", "ttpg")

for (file in files.pvar) {
    df <- read.table(paste0("gwas_results/gwas_grm/gasAcu.plink.", file, ".pvar"),
        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(chrom = CHR, pos = POS, variant.id = ID, )

    assign(paste0(file, ".pvar"), df, envir = .GlobalEnv)
}

ray.pvar <- read.table("gwas_results/morphology/plink_out/gasAcu.plink.ray.pvar",
    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(chrom = CHR, pos = POS, variant.id = ID)
```

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

```
mce.sig.ann <- mce.sig %>%
    inner_join(maxCranElev.pvar, by = c("chrom", "pos", "variant.id")) %>%
    dplyr::select(variant.id) %>%
    write_delim(., "gwas_results/gwas_grm/mce_sig_snps.txt", delim = "\t")

md.sig.ann <- md.sig %>%
    inner_join(maxDecel.pvar, by = c("chrom", "pos", "variant.id")) %>%
    dplyr::select(variant.id) %>%
    write_delim(., "gwas_results/gwas_grm/md_sig_snps.txt", delim = "\t")

time.hdvmg.sig.ann <- time.hdvmg.sugg.sig %>%
    inner_join(time_HDvMG.pvar, by = c("chrom", "pos", "variant.id")) %>%
    dplyr::select(variant.id) %>%
    write_delim(., "gwas_results/gwas_grm/time_hdvmg_sugg_sig_snps.txt", delim = "\t")

time.mdvmg.sig.ann <- time.mdvmg.sugg.sig %>%
    inner_join(time_HDvMG.pvar, by = c("chrom", "pos", "variant.id")) %>%
    dplyr::select(variant.id) %>%
    write_delim(., "gwas_results/gwas_grm/time_mdvmg_sugg_sig_snps.txt", delim = "\t")

ttpg.sig.ann <- ttpg.sig %>%
    inner_join(ttpg.pvar, by = c("chrom", "pos", "variant.id")) %>%
    dplyr::select(variant.id) %>%
    write_delim(., "gwas_results/gwas_grm/ttpg_sig_snps.txt", delim = "\t")
```

```
ray.sig.ann <- ray.sig %>%
    inner_join(ray.pvar, by = c("chrom", "pos", "variant.id")) %>%
    dplyr::select(variant.id) %>%
    write_delim(., "gwas_results/morphology/gwas_out/ray_sig_snps.txt", delim = "\t")
```

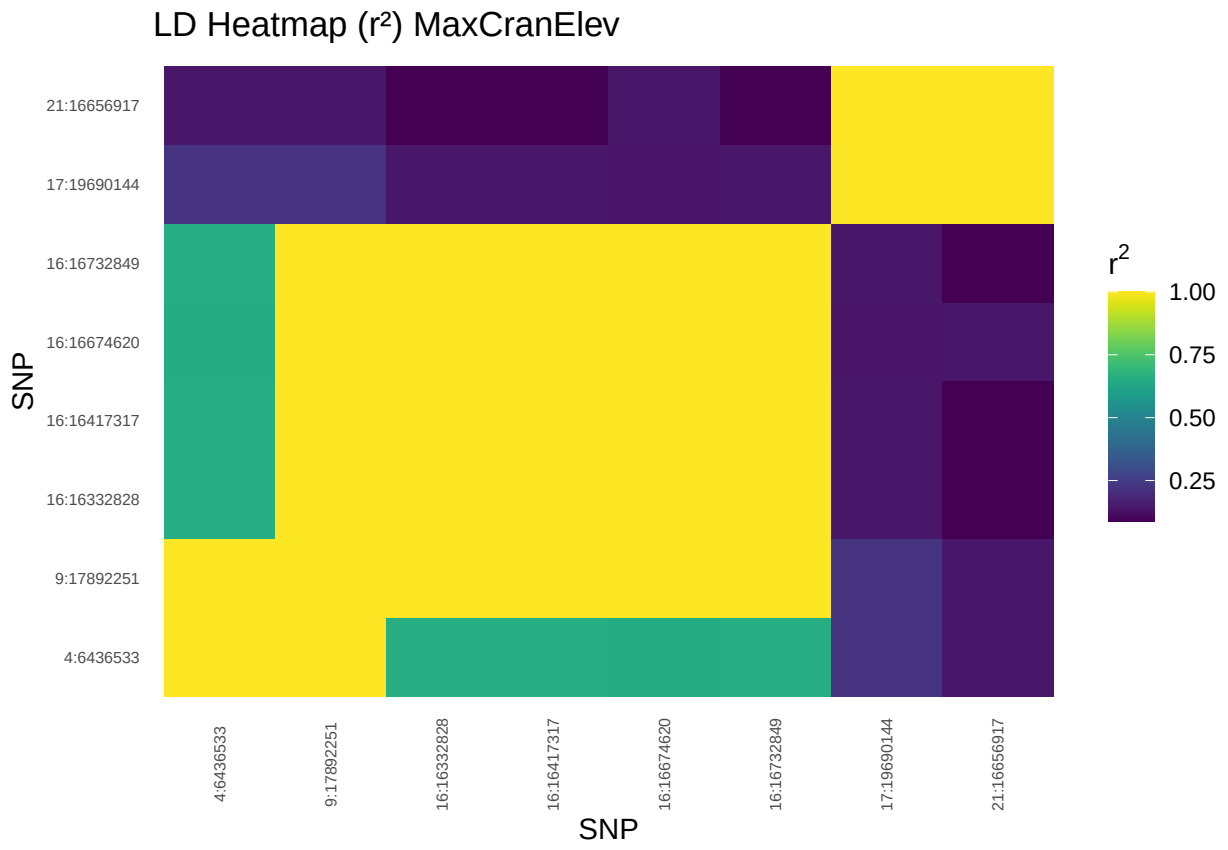
LD Matrix Generation

```
conda activate plink2

plink --bfile gasAcu.plink19.maxCranElev --extract mce_sig_snps.txt --r2 square --allow-extra-chr --out
plink --bfile gasAcu.plink19.maxDecel --extract md_sig_snps.txt --r2 square --allow-extra-chr --out md_
plink --bfile gasAcu.plink19.time_HDvMG --extract time_hdvmg_sugg_sig_snps.txt --r2 square --allow-extra
plink --bfile gasAcu.plink19.time_maxDecelvMG --extract time_mdvmg_sugg_sig_snps.txt --r2 square --allo
plink --bfile gasAcu.plink19.ttpg --extract ttpg_sig_snps.txt --r2 square --allow-extra-chr --out ttpg_
```

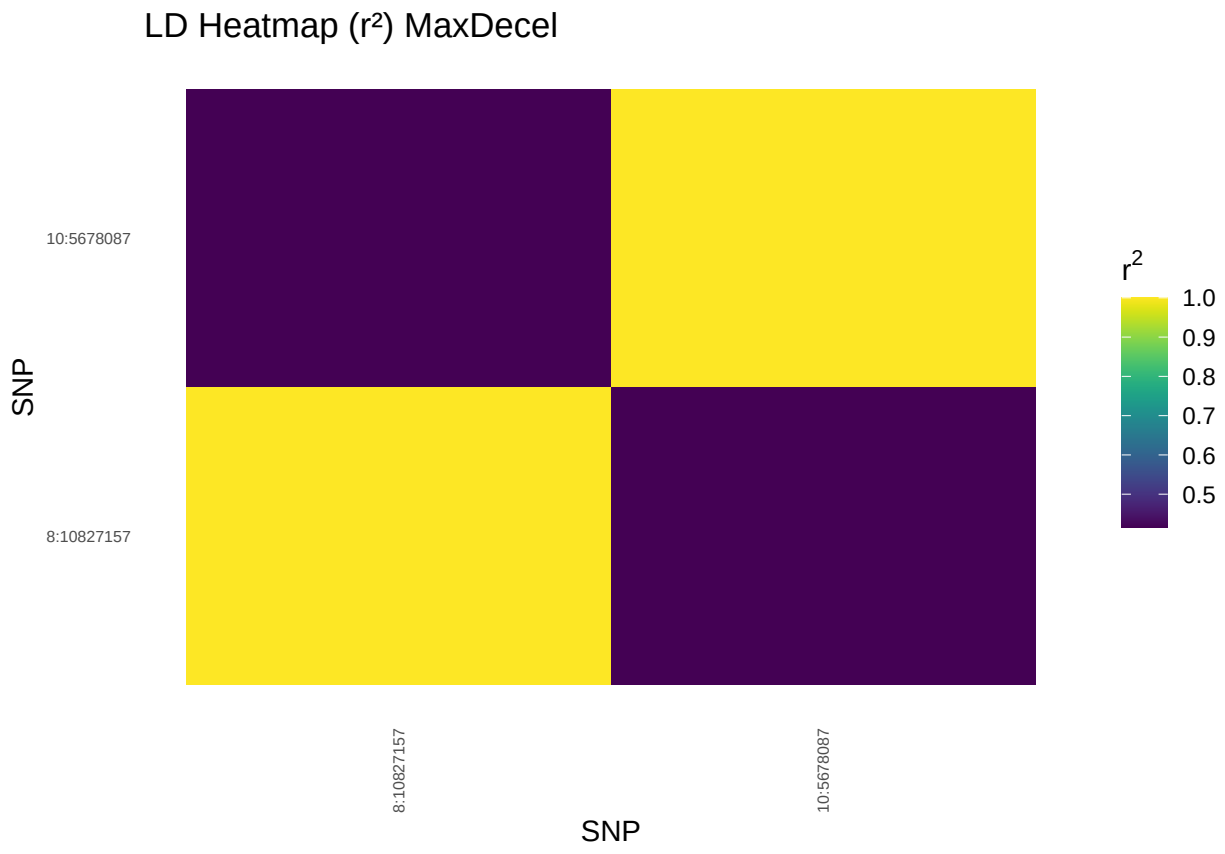
LD visualizations

```
mce.ld.mat <- read.table("gwas_results/gwas_grm/mce_ld_matrix.ld", header = F)
rownames(mce.ld.mat) <- mce.sig.ann$variant.id
colnames(mce.ld.mat) <- mce.sig.ann$variant.id

mce.ld.long <- melt(as.matrix(mce.ld.mat), varnames = c("SNP_A", "SNP_B"), value.name = "R2")

md.ld.mat <- read.table("gwas_results/gwas_grm/md_ld_matrix.ld", header = F)
rownames(md.ld.mat) <- md.sig.ann$variant.id
colnames(md.ld.mat) <- md.sig.ann$variant.id

md.ld.long <- melt(as.matrix(md.ld.mat), varnames = c("SNP_A", "SNP_B"), value.name = "R2")

time_hdvmg.ld.mat <- read.table("gwas_results/gwas_grm/time_hdvmg_ld_matrix.ld",
    header = F)
rownames(time_hdvmg.ld.mat) <- time.hdvmg.sig.ann$variant.id
colnames(time_hdvmg.ld.mat) <- time.hdvmg.sig.ann$variant.id

time_hdvmg.ld.long <- melt(as.matrix(time_hdvmg.ld.mat), varnames = c("SNP_A", "SNP_B"),
    value.name = "R2")

time_mdvmg.ld.mat <- read.table("gwas_results/gwas_grm/time_mdvmg_ld_matrix.ld",
    header = F)
rownames(time_mdvmg.ld.mat) <- time.mdvmg.sig.ann$variant.id
colnames(time_mdvmg.ld.mat) <- time.mdvmg.sig.ann$variant.id

time_mdvmg.ld.long <- melt(as.matrix(time_mdvmg.ld.mat), varnames = c("SNP_A", "SNP_B"),
    value.name = "R2")

ttpg.ld.mat <- read.table("gwas_results/gwas_grm/ttpg_ld_matrix.ld", header = F)
rownames(ttpg.ld.mat) <- ttpg.sig.ann$variant.id
colnames(ttpg.ld.mat) <- ttpg.sig.ann$variant.id
```

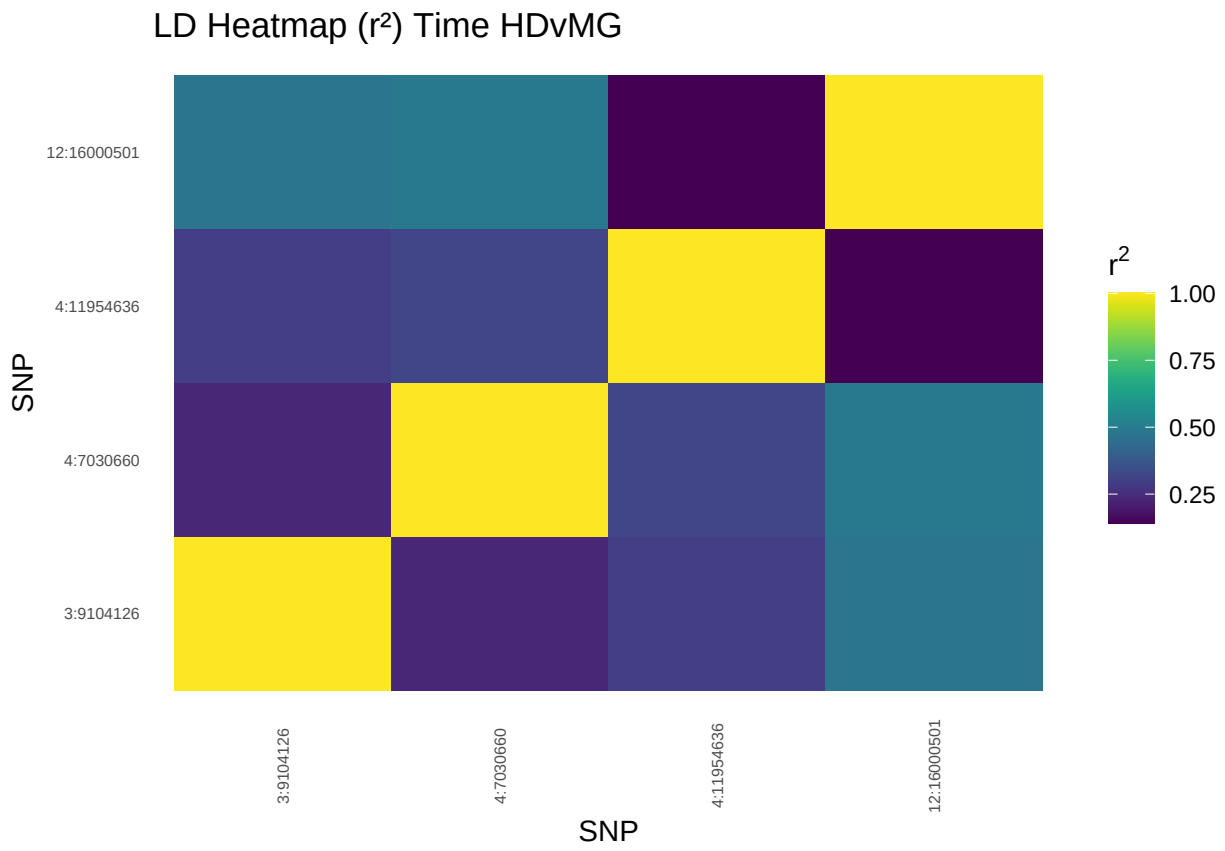
```
ttpg.ld.long <- melt(as.matrix(ttpg.ld.mat), varnames = c("SNP_A", "SNP_B"), value.name = "R2")

ray.ld.mat <- read.table("gwas_results/morphology/gwas_out/ray_ld_matrix.ld", header = F)
rownames(ray.ld.mat) <- ray.sig.ann$variant.id
colnames(ray.ld.mat) <- ray.sig.ann$variant.id

ray.ld.long <- melt(as.matrix(ray.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_viridis(option = "
    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)
    x = "SNP", y = "SNP", fill = expression(r^2))
```



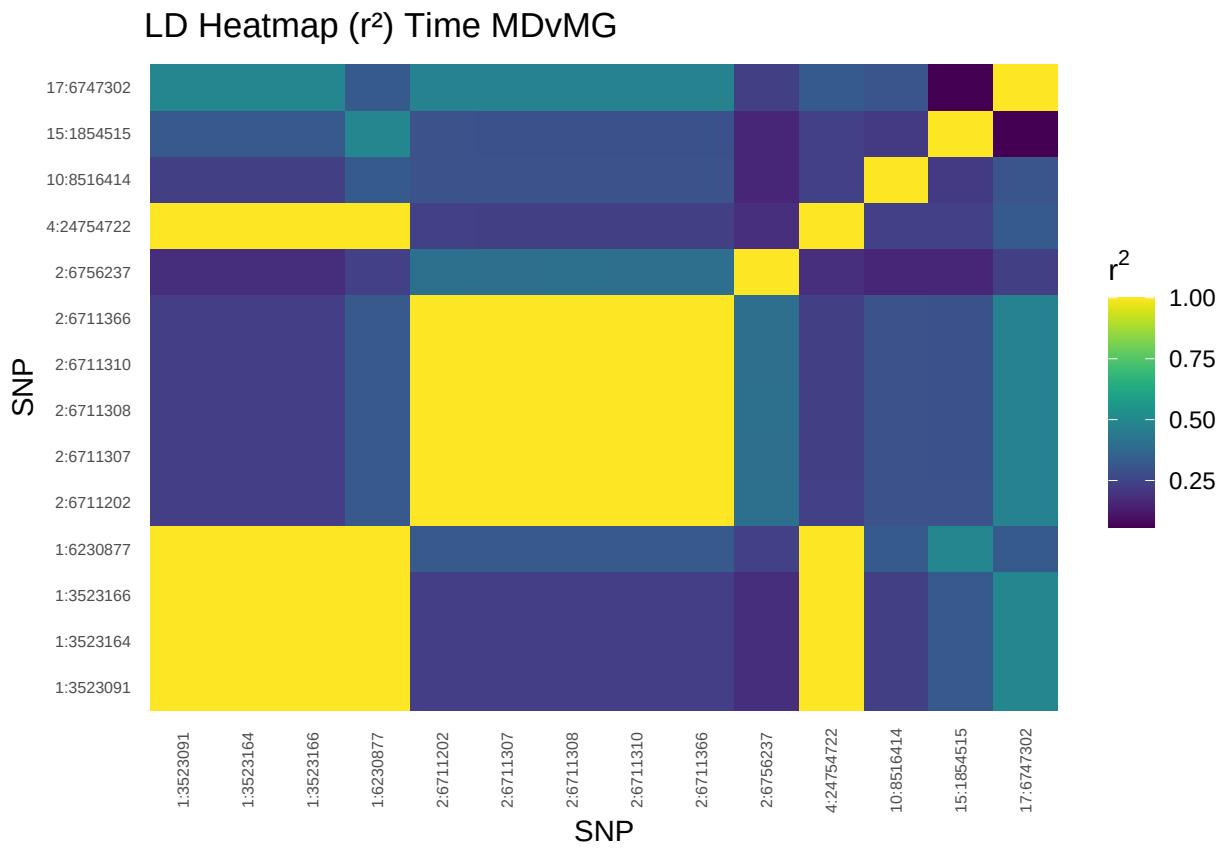

```
ggplot(md.ld.long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() + scale_fill_viridis(option = "v
    theme_minimal() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 6),
    axis.text.y = element_text(size = 6), panel.grid = element_blank()) + labs(title = "LD Heatmap (r²)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



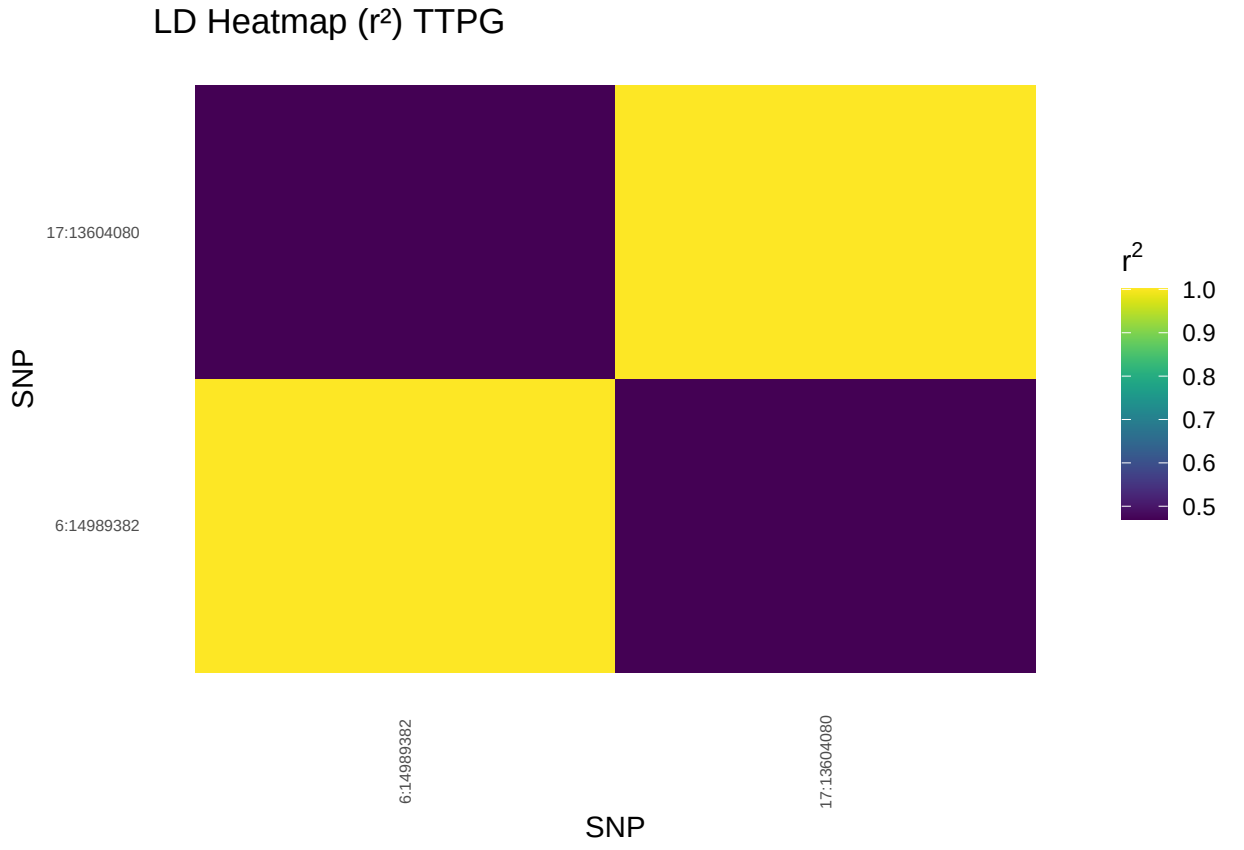

```
ggplot(time_hdvmg.ld.long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() +
    scale_fill_viridis(option = "viridis") + theme_minimal() + theme(axis.text.x = element_text(angle =
    vjust = 0.5, size = 6), axis.text.y = element_text(size = 6), panel.grid = element_blank()) +
    labs(title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



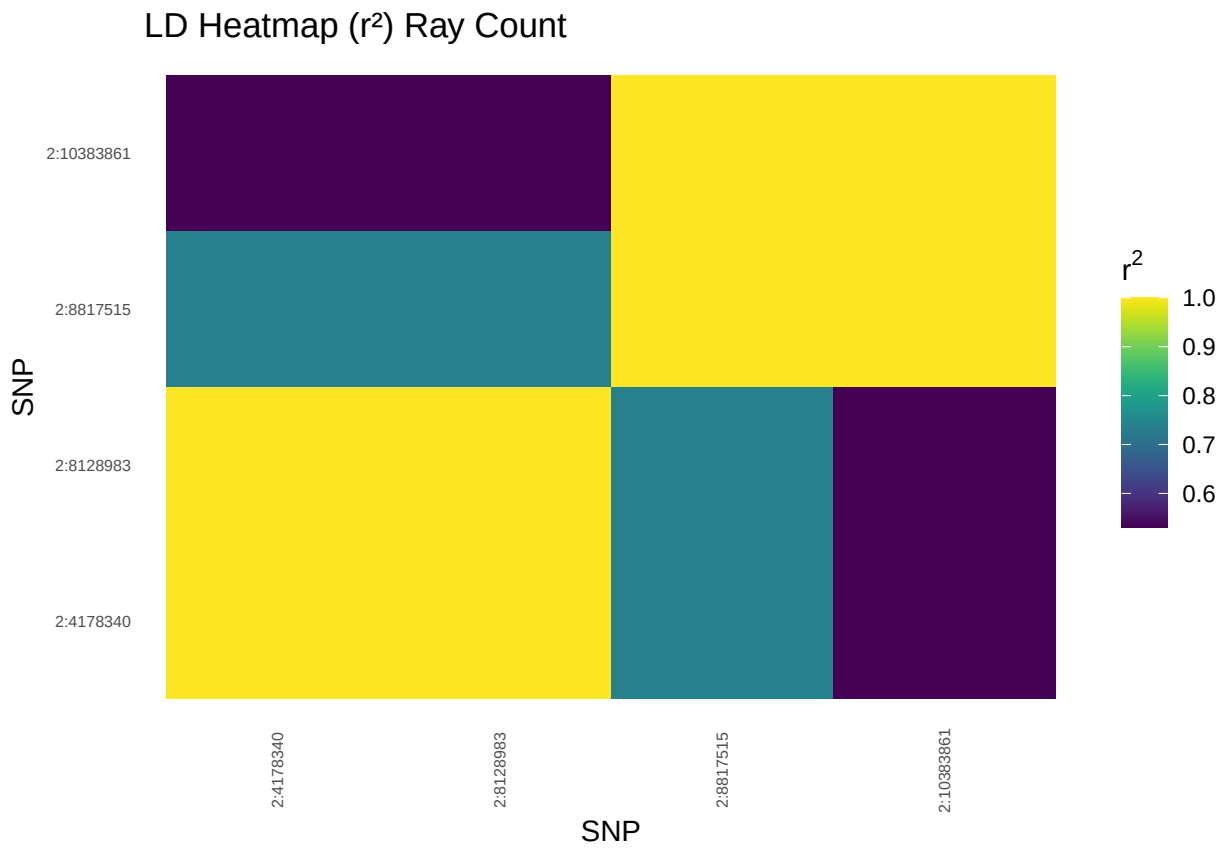

```
ggplot(time_mdvmg.ld.long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() +
    scale_fill_viridis(option = "viridis") + theme_minimal() + theme(axis.text.x = element_text(angle =
    vjust = 0.5, size = 6), axis.text.y = element_text(size = 6), panel.grid = element_blank()) +
    labs(title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




```
ggplot(ttpg.ld.long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() + scale_fill_viridis(option =
    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)
    x = "SNP", y = "SNP", fill = expression(r^2))
```




```
ggplot(ray.ld.long, aes(x = SNP_A, y = SNP_B, fill = R2)) + geom_tile() + scale_fill_viridis(option = "
    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)
    x = "SNP", y = "SNP", fill = expression(r^2))
```



SNP Gene Association

```
#### Make .BED for gene association
mce_bed <- mce.sig %>%
    inner_join(maxCranElev.pvar, by = c("chrom", "pos", "variant.id")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "mce") %>%
    select(chrom, start, end, variant.id, pheno)

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

thdvmg_bed <- time.hdvmg.sugg.sig %>%
    inner_join(time_HDvMG.pvar, by = c("chrom", "pos", "variant.id")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "thdvmg") %>%
    select(chrom, start, end, variant.id, pheno)
```

```
tmdvmg_bed <- time.mdvmg.sugg.sig %>%
    inner_join(time_HDvMG.pvar, by = c("chrom", "pos", "variant.id")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "tmdvmg") %>%
    select(chrom, start, end, variant.id, pheno)

ttpg_bed <- ttpg.sig %>%
    inner_join(ttpg.pvar, by = c("chrom", "pos", "variant.id")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "ttpg") %>%
    select(chrom, start, end, variant.id, pheno)

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

ray_bed <- ray.sig %>%
    inner_join(ray.pvar, by = c("chrom", "pos", "variant.id")) %>%
    select(-c(REF, ALT, p)) %>%
    rename(end = pos) %>%
    mutate(start = end - 1, pheno = "ray") %>%
    select(chrom, start, end, variant.id, pheno)

write_delim(ray_bed, "gwas_results/morphology/gwas_out/rayGenes.bed", delim = "\t",
    col_names = F)
```

```
# in ~/projects/def-sjsmith/sjsmith/stickles_ucr/
conda activate bedtools

mkdir geneAssoc
cp reference/GCF_016920845.1/gasAcu.gff eference/GCF_016920845.1/gasAcu_acckey geneAssoc/
cd geneAssoc/

# convert GFF to BED & pull out only CDS regions
awk '$3 == "CDS"' gasAcu.gff > gasAcu.onlyCDS.gff
awk -f gff2bed.awk gasAcu.onlyCDS.gff > gasAcu.onlyCDS.bed
cat gasAcu.onlyCDS.bed | python genenames.py > gasAcu.onlyCDS.genes.bed

## make chrom sizes file for window creation
wget http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/faToTwoBit
chmod +x ./faToTwoBit

wget http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/twoBitInfo
chmod +x ./twoBitInfo

./faToTwoBit ../reference/GCF_016920845.1/GCF_016920845.1_GAculeatus_UGA_version5_genomic.fna gasAcu.fa
./twoBitInfo gasAcu.fa.2bit stdout | sort -k2rn > gasAcu.chrom.sizes

# create windows
bedtools slop -i gasAcu.onlyCDS.genes.bed -g gasAcu.chrom.sizes -b 100000 | bedtools sort -i - > gasAcu
```

```
# replace NCBI chromosome names with linkage group numbers (and MT instead of NA)
sed 's/NA/MT/g' gasAcu_acckey > gasAcu.acckey
./replace_chrs.pl gasAcu.acckey gasAcu.windows.bed > gasAcu.windows.repl.bed

# intersect clean genes BED with windows
bedtools sort -i cleanGenes.bed | bedtools intersect -a gasAcu.windows.repl.bed -b - -wb | bedtools sor
bedtools sort -i rayGenes.bed | bedtools intersect -a gasAcu.windows.repl.bed -b - -wb | bedtools sort

# clean up gene names field by removing unnamed LOCs
sed 's/LOC[0-9]\+,//g' gasAcu.genes.intersect.bed | sed 's/','/ /g' > gasAcu.genes.clean.bed
sed 's/LOC[0-9]\+,//g' gasAcu.rayGenes.intersect.bed | sed 's/','/ /g' > gasAcu.rayGenes.clean.bed
```

Create gene lists

Kinematic traits

```
genes <- read.table("gwas_results/gwas_grm/geneAssoc/gasAcu.genes.clean.bed", sep = '\t',
                    col.names = c("chr", "start", "end", "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 != "12:16000501") %>% # gets rid of one leftover LOC*
  select(-c(genes))

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

snpList <- gene_clean %>%
  select(-c(snp, pheno)) %>%
  pivot_longer(cols = everything(), values_to = "gene") %>%
  filter(!is.na(gene), gene != "") %>%
  distinct(gene)

write_delim(snpList, "gwas_results/gwas_grm/geneAssoc/snpList", delim = "\t", col_names = F)

## PANTHER interlude

list <- read_excel("gwas_results/gwas_grm/geneAssoc/geneAssoc.xlsx")

panther <- read_delim("gwas_results/gwas_grm/geneAssoc/pantherGeneList_full.txt", col_names = F) %>%
  rename(gene = "X2", sum = "X3", ref = "X6") %>%
  select(gene, sum, ref) %>%
  separate(sum, into = c("summary", NA), sep = ';', extra = "drop")

cleanList <- full_join(list, panther, by = "gene", relationship = "many-to-many") %>%
  mutate(ref = if_else(is.na(ref), "Homo sapiens", ref))
write_delim(cleanList, "gwas_results/gwas_grm/geneAssoc/table_geneAssoc.tsv", delim = "\t")
```

Morphological traits

```
genes <- read.table("gwas_results/morphology/gwas_out/gasAcu.rayGenes.clean.bed",
    sep = "\t", col.names = c("chr", "start", "end", "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) %>%
    select(-c(genes))

write_delim(gene_clean, "gwas_results/morphology/gwas_out/gasAcu.final.rayGenes.bed",
    delim = "\t")
write_delim(gene_clean, "gwas_results/morphology/gwas_out/gasAcu.final.rayGenes.txt",
    delim = "\t")
write_delim(gene_clean, "gwas_results/morphology/gwas_out/gasAcu.final.rayGenes.csv",
    delim = ",")

snpList <- gene_clean %>%
    select(-c(snp, pheno)) %>%
    pivot_longer(cols = everything(), values_to = "gene") %>%
    filter(!is.na(gene), gene != "") %>%
    distinct(gene)

write_delim(snpList, "gwas_results/morphology/gwas_out/snpList", delim = "\t", col_names = F)

## PANTHER interlude

list <- read_excel("gwas_results/morphology/gwas_out/rayGeneAssoc.xlsx")

panther <- read_delim("gwas_results/morphology/gwas_out/pantherGeneList.txt", col_names = F) %>%
    rename(gene = "X2", sum = "X3", ref = "X6") %>%
    select(gene, sum, ref) %>%
    separate(sum, into = c("summary", NA), sep = ";", extra = "drop")

cleanList <- full_join(list, panther, by = "gene", relationship = "many-to-many") %>%
    mutate(ref = if_else(is.na(ref), "Homo sapiens", ref))
write_delim(cleanList, "gwas_results/morphology/gwas_out/table_rayGeneAssoc.tsv",
    delim = "\t")
```