QTL-Sorghum

Michael Hall

3/2/2022

```
devtools::install_github("PBGLMichaelHall/QTLseqr",force = TRUE)
```

```
Downloading GitHub repo PBGLMichaelHall/QTLseqr@HEAD
```

```
* checking for file ‘/tmp/RtmpzeWI2R/remotes2674ba0e0e6/PBGLMichaelHall-QTLseqr-527dcc5/DESCRIPTION’ ..
* preparing ‘QTLseqr’:
* checking DESCRIPTION meta-information ... OK
* cleaning src
Warning: /tmp/RtmpFIvWzy/Rbuild2b5447a80ff/QTLseqr/man/tricube_Smooth.Rd:2: unexpected '}'
Warning: /tmp/RtmpFIvWzy/Rbuild2b5447a80ff/QTLseqr/man/tricube_Smooth.Rd:3: unexpected '}'
* checking for LF line-endings in source and make files and shell scripts
* checking for empty or unneeded directories
Omitted ‘LazyData’ from DESCRIPTION
* building ‘QTLseqr_0.7.5.2.tar.gz’
```

```
Installing package into '/home/michael/R/x86_64-pc-linux-gnu-library/4.1'
(as 'lib' is unspecified)
```

```
library(QTLseqr)
library(tinytex)
library(vcfR)
```

```
   *****       ***   vcfR   ***       *****
   This is vcfR 1.12.0
     browseVignettes('vcfR') # Documentation
     citation('vcfR') # Citation
   *****       *****      *****       *****
```

```
library(tidyr)
library(ggplot2)
```

```
#Set Working Directory
setwd("/home/michael/Desktop/QTLseqr/extdata")

#vcf file must only contain bialleleic variants. (filter upstream, e.g., with bcftools view -m2 -M2), a
vcf <- read.vcfR(file = "freebayes_D2.filtered.vcf")
```

```
Scanning file to determine attributes.
File attributes:
  meta lines: 937
  header_line: 938
  variant count: 7861
  column count: 13
Meta line 937 read in.
All meta lines processed.
```

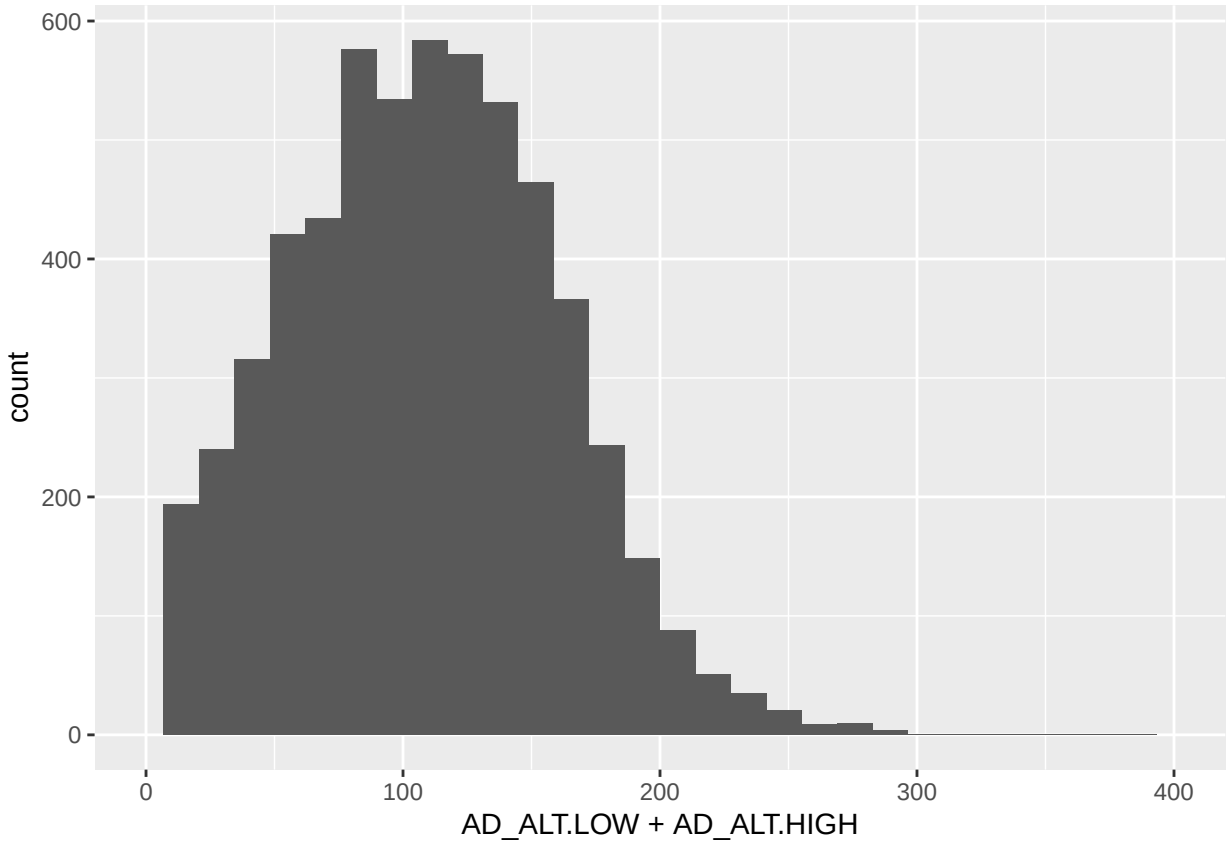
```
gt matrix initialized.
Character matrix gt created.
  Character matrix gt rows: 7861
  Character matrix gt cols: 13
  skip: 0
  nrows: 7861
  row_num: 0
Processed variant 1000Processed variant 2000Processed variant 3000Processed variant 4000Processed varia
All variants processed
```

```
#Convert to tidy data frame
VCF_TIDY <- vcfR2tidy(vcf)

#Call the Parser
QTLParser_1_MH(vcf = VCF_TIDY, HighBulk = "D2_F2_tt",LowBulk = "D2_F2_TT")
```

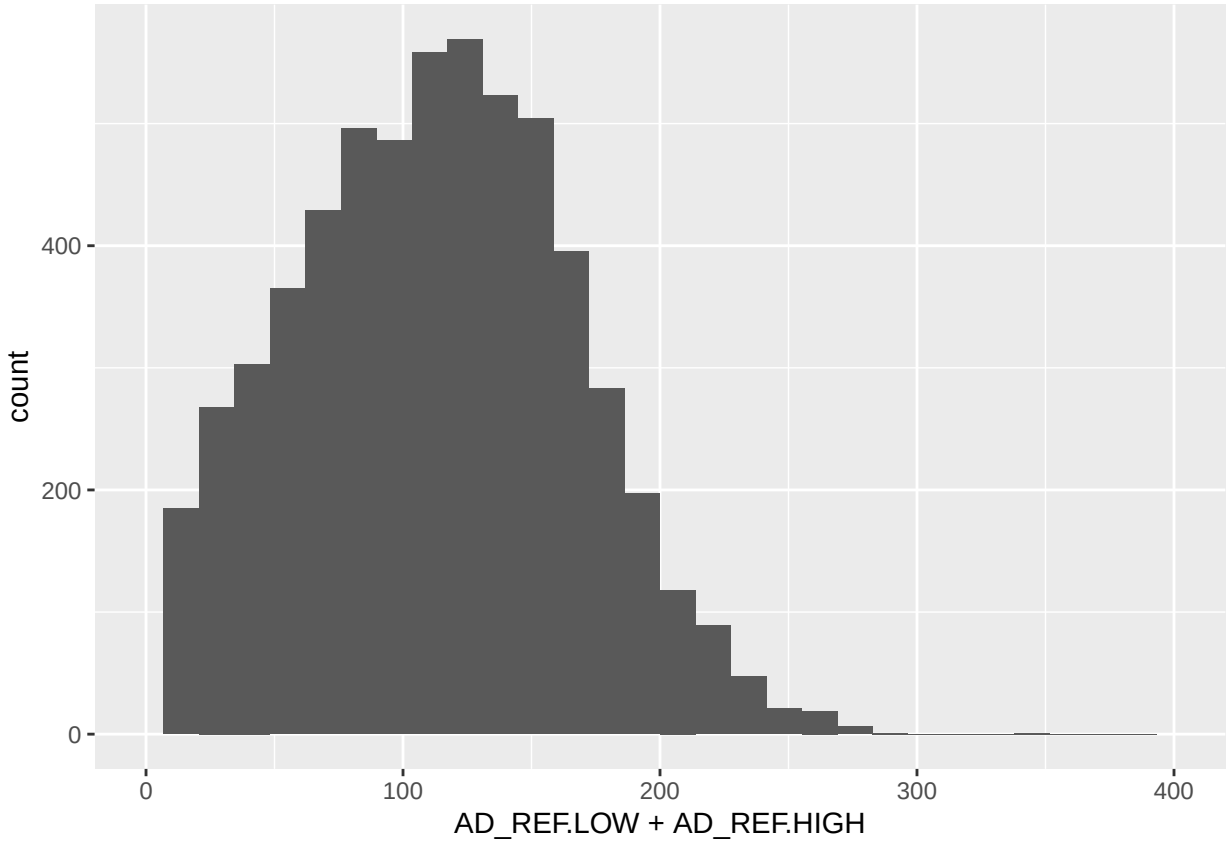
```
'data.frame':   31424 obs. of  7 variables:
 $ CHROM  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ POS    : int  344698 2943267 3751995 4720049 5567202 6237654 6582529 7047748 8720466 8720551 ...
 $ REF    : chr  "C" "T" "T" "G" ...
 $ ALT    : chr  "T" "A" "C" "A" ...
 $ DP     : int  6 30 8 30 22 10 33 1 3 1 ...
 $ var1   : chr  "14,23" "66,51" "15,10" "80,37" ...
 $ Samples: chr  "con-all" "con-all" "con-all" "con-all" ...
'data.frame':   31400 obs. of  7 variables:
 $ CHROM  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ POS    : int  344698 2943267 3751995 4720049 5567202 6237654 6582529 7047748 8720466 8720551 ...
 $ REF    : chr  "C" "T" "T" "G" ...
 $ ALT    : chr  "T" "A" "C" "A" ...
 $ DP     : int  6 30 8 30 22 10 33 1 3 1 ...
 $ var1   : chr  "19,18" "44,42" "8,4" "64,50" ...
 $ Samples: chr  "con-all" "con-all" "con-all" "con-all" ...
```

```
#Set High bulk and Low bulk sample names and parser generated file name

HighBulk <- "D2_F2_tt"
LowBulk <- "D2_F2_TT"
file <- "Hall.csv"

#Choose which chromosomes will be included in the analysis,
#the tidy data frame makes a CHROMKEY so no need to change chromosome names
Chroms <- 1:10


df <-
  importFromTable(
    file = file,
    highBulk = HighBulk,
    lowBulk = LowBulk,
    chromList = Chroms
  )


#plot histograms associated with filtering arguments to determine if cut off values are appropriate
```

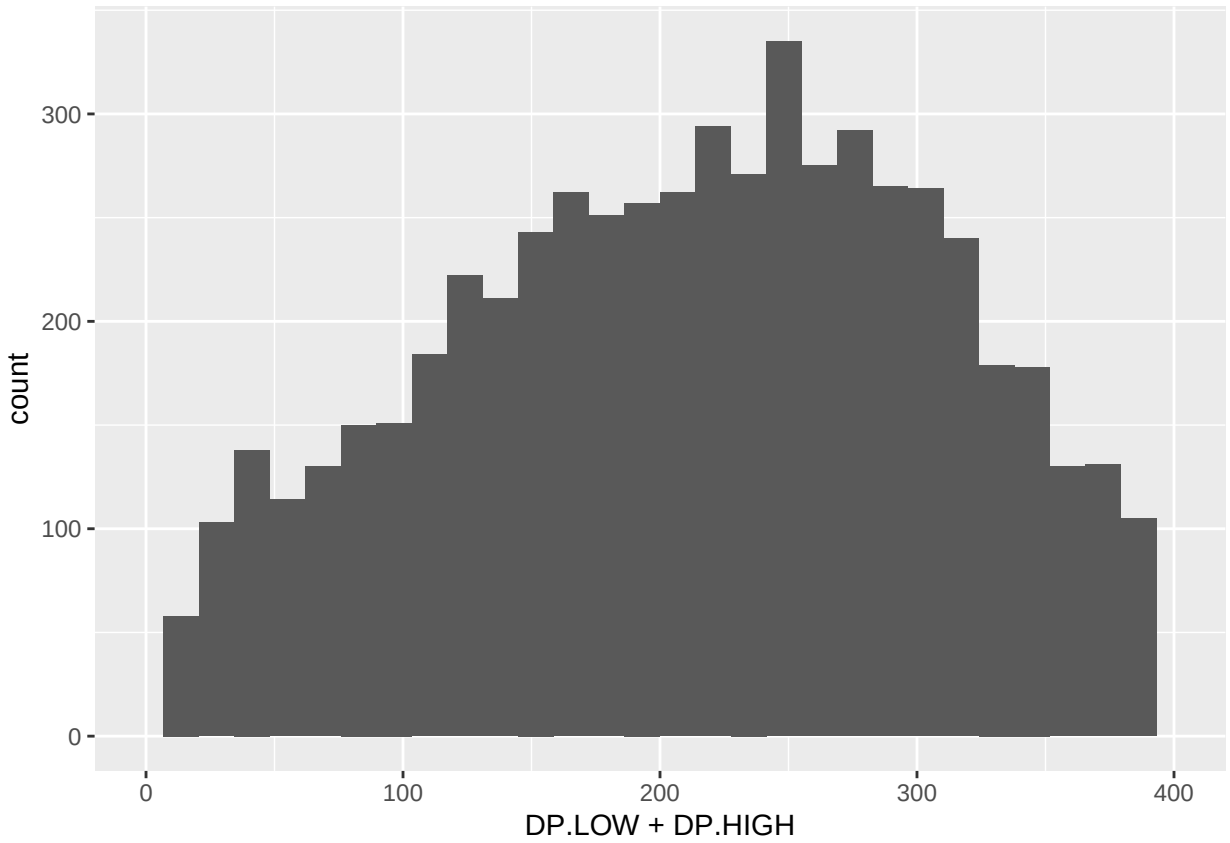
```
ggplot(data = df) +
  geom_histogram(aes(x = AD_ALT.LOW + AD_ALT.HIGH)) + xlim(0,400)
```



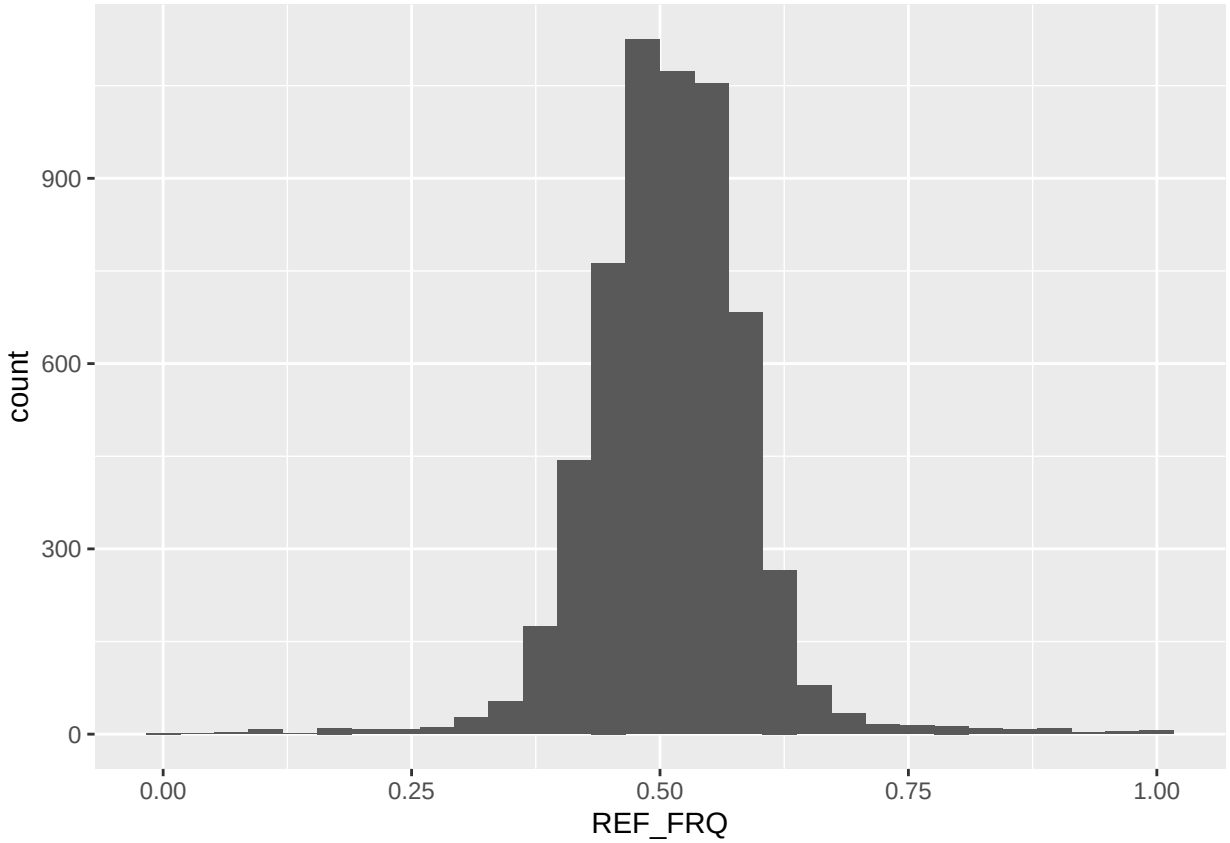

```
ggsave(filename = "AD_Histogram.png",plot = last_plot())
ggplot(data = df) +
  geom_histogram(aes(x = AD_REF.LOW + AD_REF.HIGH)) + xlim(0,400)
```




```
ggsave(filename = "AD_Ref_Histogram.png",plot = last_plot())
ggplot(data =df) +
  geom_histogram(aes(x = DP.LOW + DP.HIGH)) + xlim(0,400)
```




```
ggsave(filename = "Depth_Histogram.png",plot=last_plot())
ggplot(data = df) +
  geom_histogram(aes(x = REF_FRQ))
```




```
ggsave(filename = "Ref_Freq_Histogram.png",plot = last_plot())
```

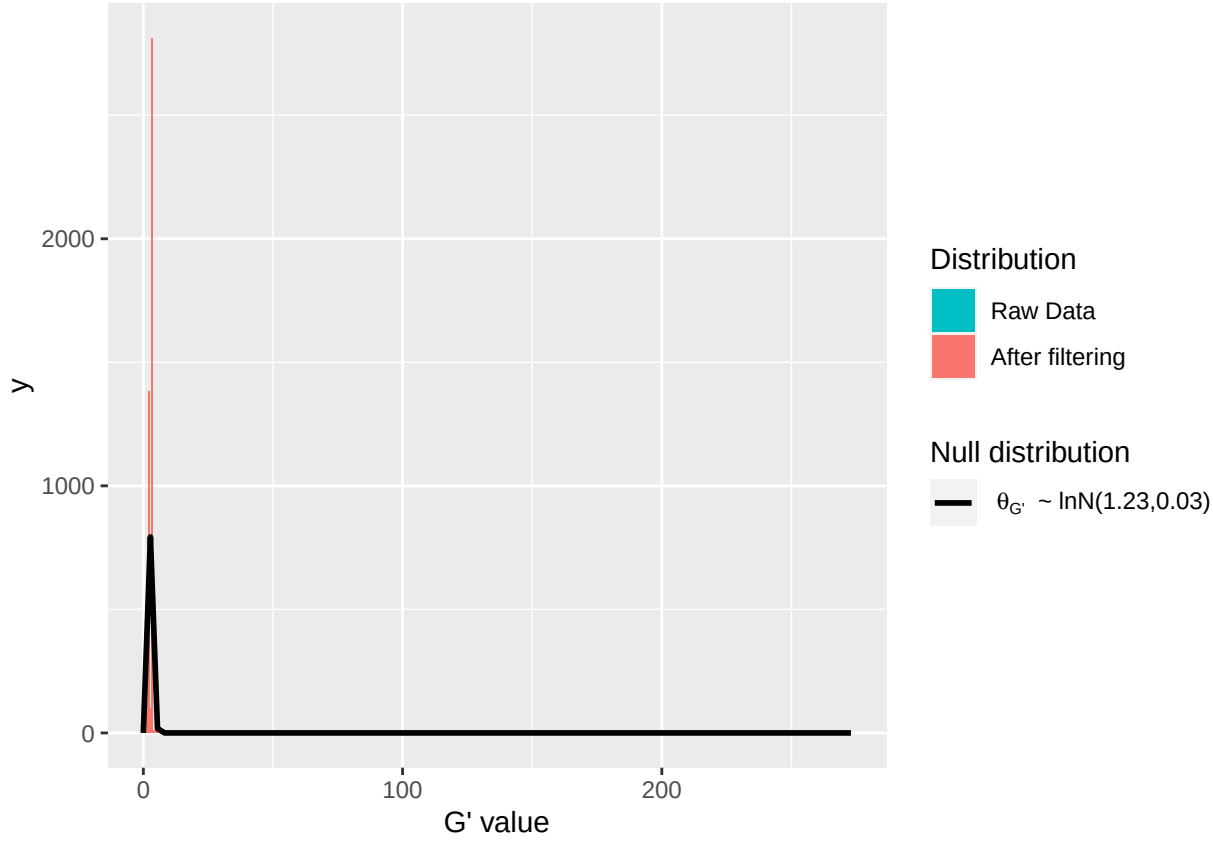
```
#Filter SNPs based on some criteria
df_filt <-
  filterSNPs(
    SNPset = df,
    refAlleleFreq = 0.20,
    minTotalDepth = 100,
    maxTotalDepth = 400,
    minSampleDepth = 40,
    #    minGQ = 0
  )
```

```
Filtering by reference allele frequency: 0.2 <= REF_FRQ <= 0.8
...Filtered 86 SNPs
Filtering by total sample read depth: Total DP >= 100
...Filtered 733 SNPs
Filtering by total sample read depth: Total DP <= 400
...Filtered 175 SNPs
Filtering by per sample read depth: DP >= 40
...Filtered 22 SNPs
Original SNP number: 5917, Filtered: 1016, Remaining: 4901
```

```
#Run G' analysis
df_filt<-runGprimeAnalysis_MH(
  SNPset = df_filt,
  windowSize = 5000000,
  outlierFilter = "deltaSNP",
  filterThreshold = 0.1)
```

```
Counting SNPs in each window...
Calculating tricube smoothed delta SNP index...
Calculating G and G' statistics...
Using deltaSNP-index to filter outlier regions with a threshold of 0.1
Estimating the mode of a trimmed G prime set using the 'modeest' package...
Calculating p-values...
```

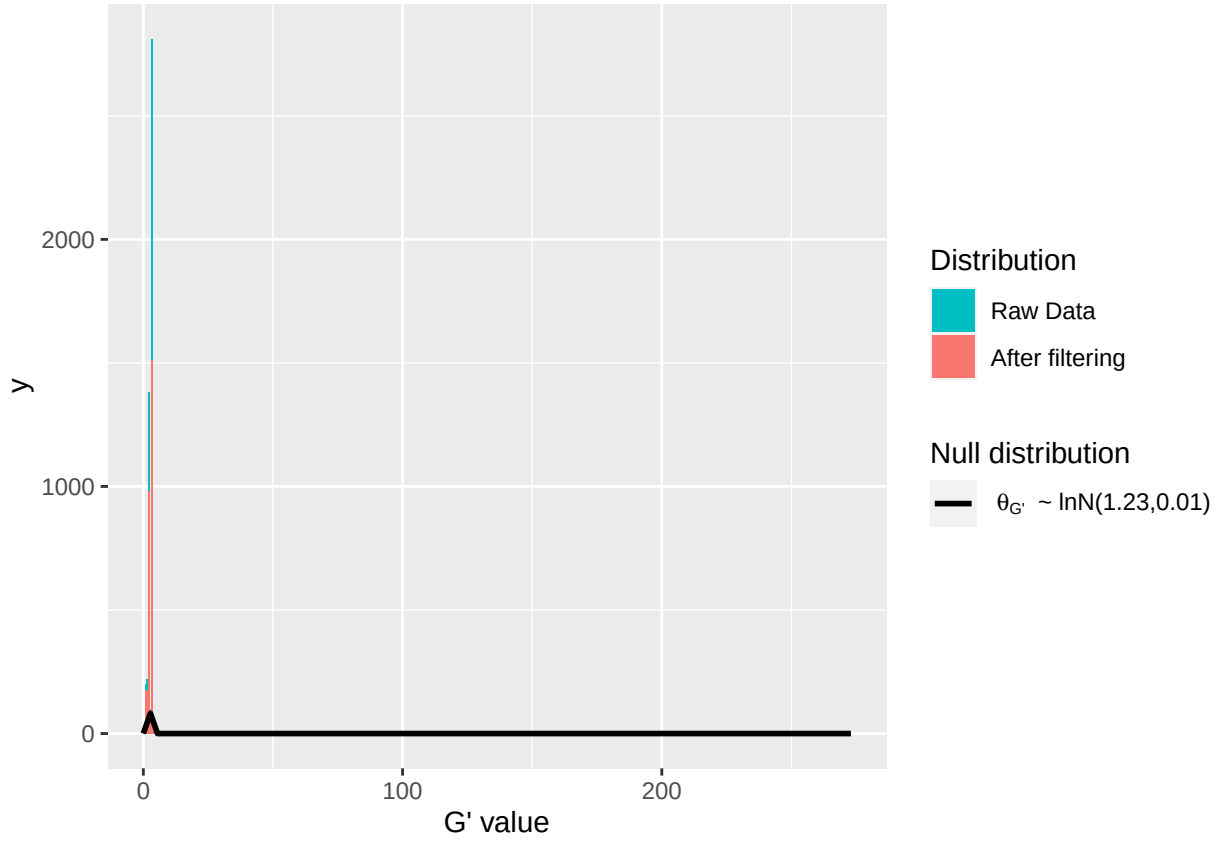
```
setwd("/home/michael/Desktop/SorghumQTL/GPrimeDistributionPlots/")



# G' Distribution Plot
plotGprimeDist(SNPset = df_filt, outlierFilter = "Hampel")
```




```
ggsave(filename = "Hampel_GPrime.png",plot = last_plot())
```

```
Saving 6.5 x 4.5 in image
```

```
setwd("/home/michael/Desktop/SorghumQTL/DeltaSNP/")
plotGprimeDist(SNPset = df_filt, outlierFilter = "deltaSNP",filterThreshold = 0.1)
```




```
ggsave(filename = "DeltaSNP.png",plot = last_plot())
```

```
Saving 6.5 x 4.5 in image
```

```
#Run QTLseq analysis
df_filt2 <- runQTLseqAnalysis_MH(
  SNPset = df_filt,
  windowSize = 5000000,
  popStruc = "F2",
  bulkSize = c(45, 38),
  replications = 10000,
  intervals = c(95, 99)
)
```

```
Counting SNPs in each window...

Calculating tricube smoothed delta SNP index...

Returning the following two sided confidence intervals: 95, 99

Variable 'depth' not defined, using min and max depth from data: 40-198

Assuming bulks selected from F2 population, with 45 and 38 individuals per bulk.

Simulating 10000 SNPs with reads at each depth: 40-198

Keeping SNPs with >= 0.3 SNP-index in both simulated bulks
```

```
Joining, by = "tricubeDP"
```

```
setwd("/home/michael/Desktop/SorghumQTL/nSNPs/")

#make the Plot
snpnumber <- plotQTLStats(SNPset = df_filt2, var = "nSNPs")
ggsave(filename = "nSNPs.png",plot = last_plot())
```

```
Saving 6.5 x 4.5 in image
```

```
setwd("/home/michael/Desktop/SorghumQTL/GPrimeDistributionPlots/")
Gprime<-plotQTLStats(SNPset = df_filt, var = "Gprime", plotThreshold = TRUE, q = 0.01)
ggsave(filename = "GPrime.png",plot = last_plot())
```

```
Saving 6.5 x 4.5 in image
```

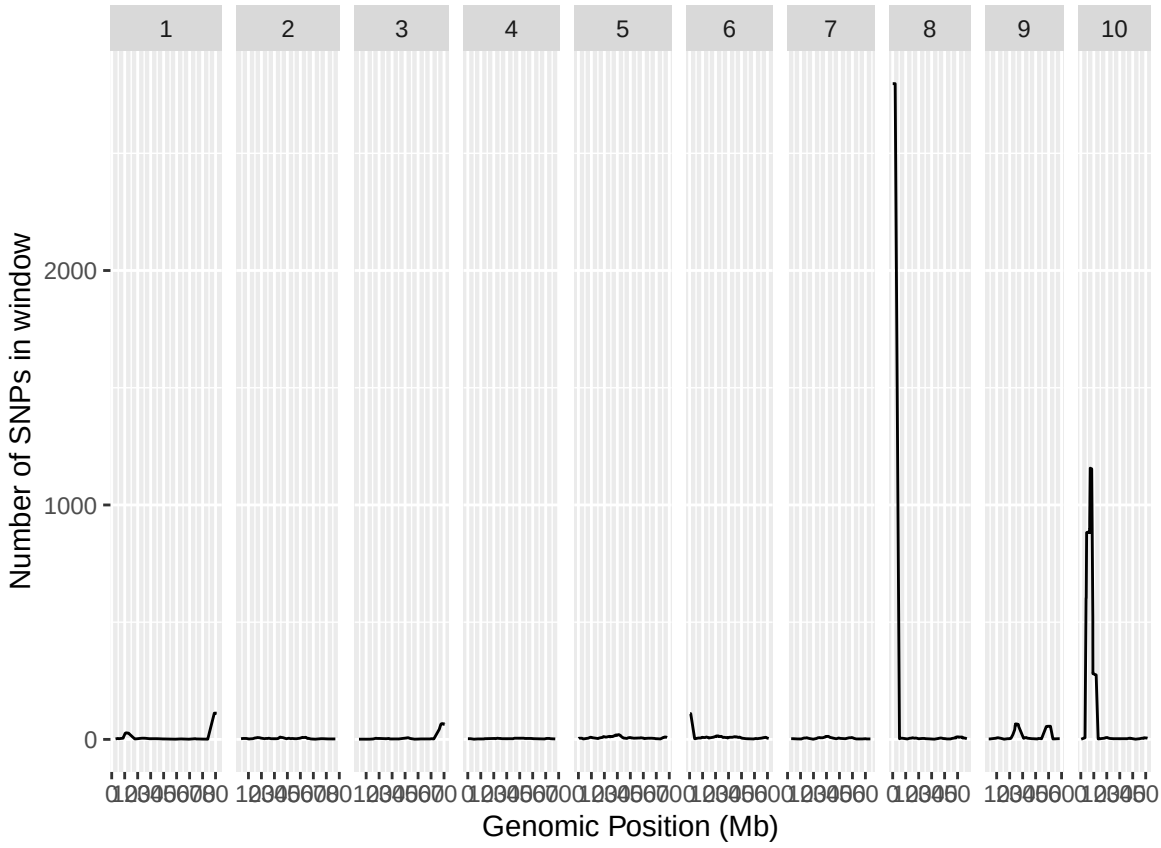
```
setwd("/home/michael/Desktop/SorghumQTL/DeltaSNP/")
deltaSNP<-plotQTLStats(SNPset = df_filt2, var = "deltaSNP", plotIntervals  = TRUE)
ggsave(filename = "DeltaSNPInterval.png",plot = last_plot())
```

```
Saving 6.5 x 4.5 in image
```

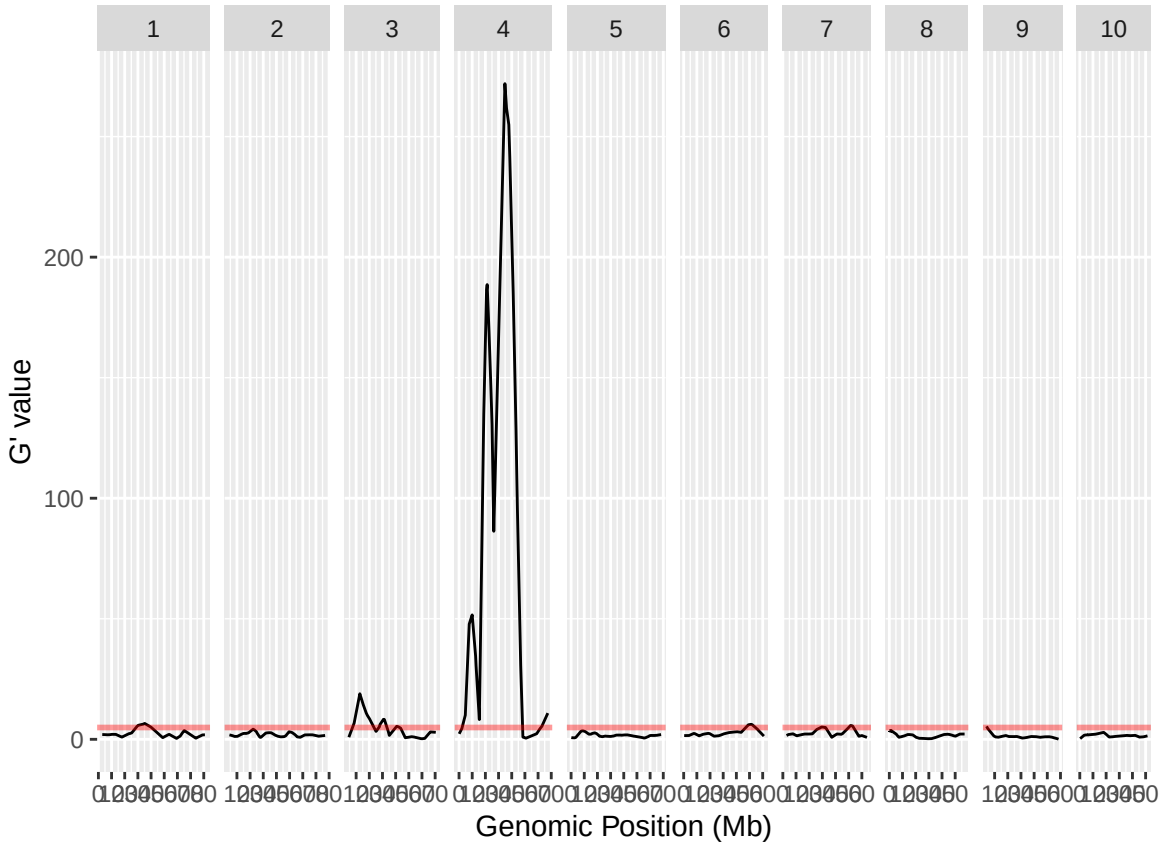
```
setwd("/home/michael/Desktop/SorghumQTL/negLog10Pval/")
neglog<-plotQTLStats(SNPset = df_filt2, var = "negLog10Pval",plotThreshold = TRUE,q=0.01,subset = c("1"
ggsave(filename = "negLog10Pval.png",plot = last_plot())
```

```
Saving 6.5 x 4.5 in image
```

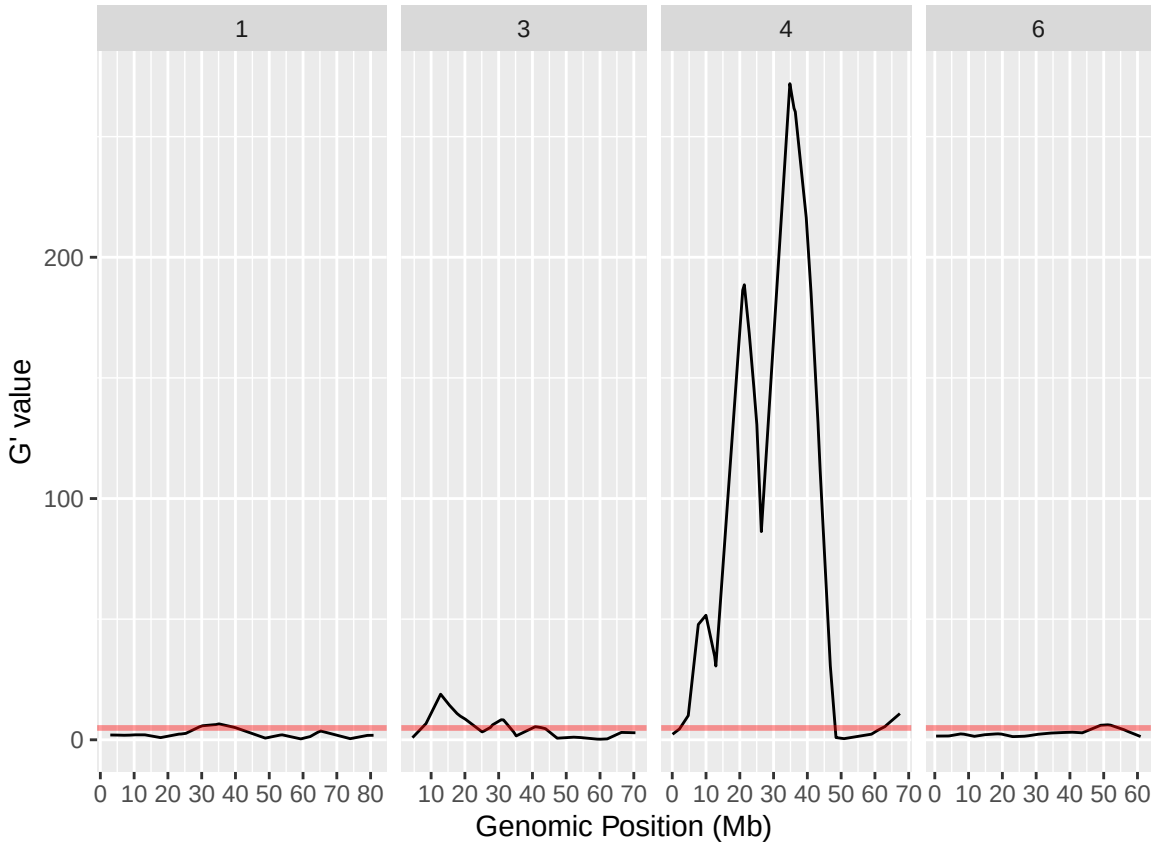
```
Gprime2<-plotQTLStats(SNPset = df_filt2, var = "Gprime",plotThreshold = TRUE,q=0.01,subset = c("1","3",
#plot the plots
snpnumber
```


1
2
3
4
5
6
7
8
9
10
Number of SNPs in window
2000
1000
0
Genomic Position (Mb)

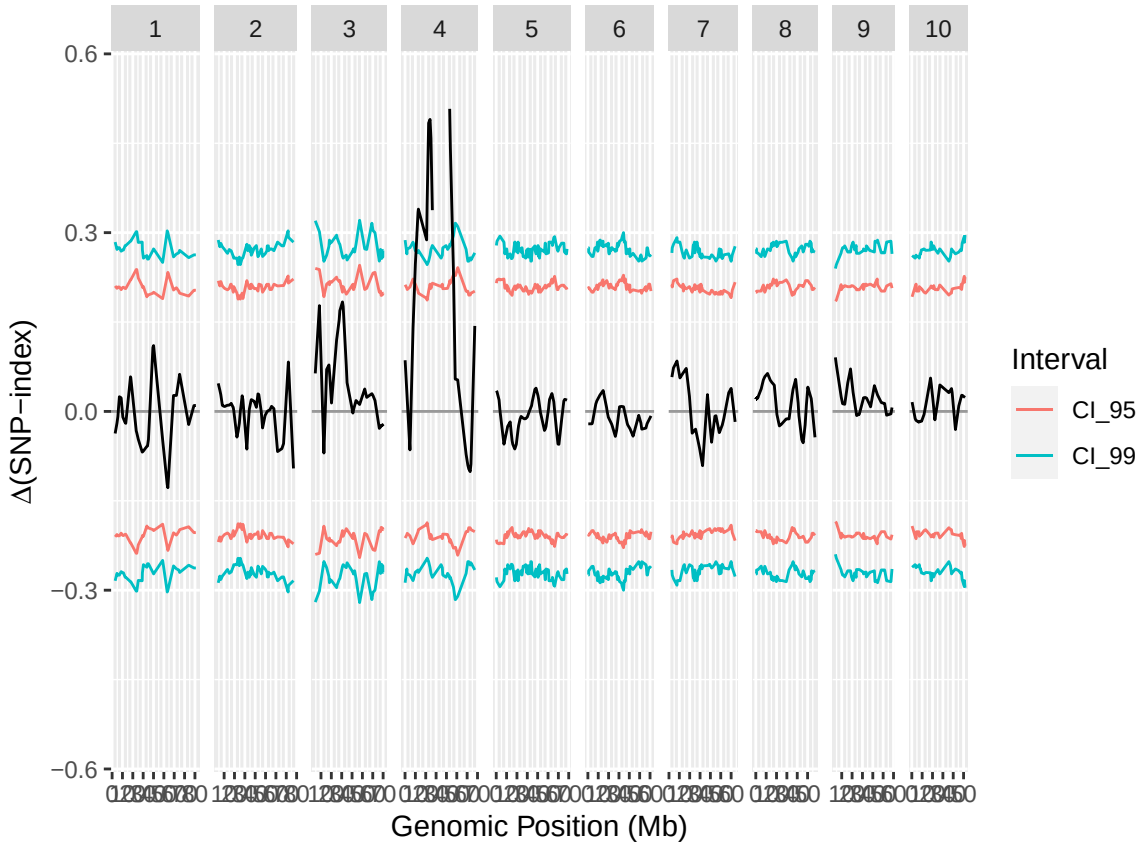
```
Gprime
```



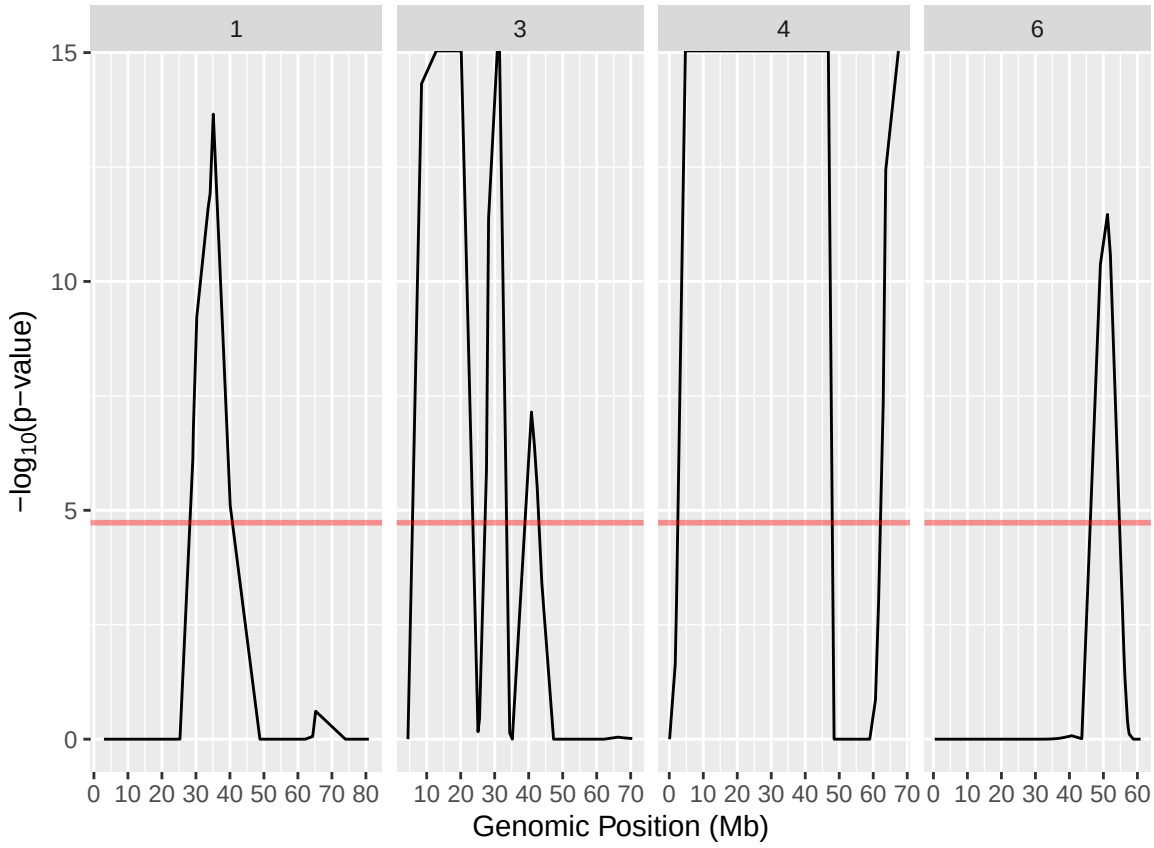

```
Gprime2
```


1
3
4
6
200
100
0
G' value
0 10 20 30 40 50 60 70 80
10 20 30 40 50 60 70
0 10 20 30 40 50 60 70
0 10 20 30 40 50 60
Genomic Position (Mb)

```
deltaSNP
```


1
2
3
4
5
6
7
8
9
10
0.6
0.3
0.0
−0.3
−0.6
Δ(SNP−index)
Genomic Position (Mb)
Interval
CI_95
CI_99

```
neglog
```




```
#export summary CSV
setwd("/home/michael/Desktop/SorghumQTL/PeakSummary/")
QTLTable <- getQTLTable(SNPset = df_filt, alpha = 0.01, export = TRUE, fileName = "my_BSA_QTL.csv")
write.csv(QTLTable, file = "QTLTablePeaks.csv", row.names = FALSE, col.names = TRUE)
Table4 <- read.table(file = "QTLTablePeaks.csv",header = TRUE, sep = ",", fill=TRUE)
```