Rice-QTL

Michael Hall

3/16/2022

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

```
Downloading GitHub repo PBGLMichaelHall/QTLseqr@HEAD

* checking for file ‘/tmp/RtmpceWZBm/remotes585218331831/PBGLMichaelHall-QTLseqr-e4dec9a/DESCRIPTION’ .
* preparing ‘QTLseqr’:
* checking DESCRIPTION meta-information ... OK
* cleaning src
Warning: /tmp/Rtmp2oSN8T/Rbuild589414baf485/QTLseqr/man/obs_M.Rd:13: unknown macro '\item'
Warning: /tmp/Rtmp2oSN8T/Rbuild589414baf485/QTLseqr/man/obs_MH.Rd:13: unknown macro '\item'
Warning: /tmp/Rtmp2oSN8T/Rbuild589414baf485/QTLseqr/man/tricube_Smooth.Rd:2: unexpected '}'
Warning: /tmp/Rtmp2oSN8T/Rbuild589414baf485/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)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(ggrepel)
library(ggpubr)
```

```r
#Set Working Directory
setwd("/home/michael/Desktop/RiceCold2")

#vcf file must only contain bialleleic variants. (filter upstream, e.g., with bcftools view -m2 -M2), a
vcf <- read.vcfR(file = "wGQ-Filt-freebayes~bwa~IRGSP-1.0~both-segregant_bulks~filtered-default.vcf.gz")
```

```
Scanning file to determine attributes.
File attributes:
  meta lines: 126
  header_line: 127
  variant count: 1714745
  column count: 11
Meta line 126 read in.
All meta lines processed.
gt matrix initialized.
Character matrix gt created.
  Character matrix gt rows: 1714745
  Character matrix gt cols: 11
  skip: 0
  nrows: 1714745
  row_num: 0
Processed variant 1000Processed variant 2000Processed variant 3000Processed variant 4000Processed varia
All variants processed
```

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

#Call the Parser
QTLParser_1_MH(vcf = VCF_TIDY, HighBulk = "ET-pool-385",LowBulk = "ES-pool-430")
```

```
'data.frame':   3429490 obs. of  7 variables:
 $ CHROM  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ POS    : int  1037 1052 23314 31071 31478 33667 34057 34339 35239 38389 ...
 $ REF    : chr  "TAAACCC" "AAACCCTAACCCTA" "GTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTA
 $ ALT    : chr  "TAACCC" "AAACCCTA" "GTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATC
 $ DP     : int  7 6 24 83 87 75 88 81 71 86 ...
 $ var1   : chr  "1,1" "0,2" "16,17" "39,36" ...
 $ Samples: chr  "ES-pool-430" "ES-pool-430" "ES-pool-430" "ES-pool-430" ...
'data.frame':   3429490 obs. of  7 variables:
 $ CHROM  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ POS    : int  1037 1052 23314 31071 31478 33667 34057 34339 35239 38389 ...
 $ REF    : chr  "TAAACCC" "AAACCCTAACCCTA" "GTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTA
 $ ALT    : chr  "TAACCC" "AAACCCTA" "GTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATC
 $ DP     : int  7 6 24 83 87 75 88 81 71 86 ...
 $ var1   : chr  "3,4" "0,5" "15,9" "36,47" ...
 $ Samples: chr  "ES-pool-430" "ES-pool-430" "ES-pool-430" "ES-pool-430" ...
```

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

HighBulk <- "ET-pool-385"
LowBulk <- "ES-pool-430"
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
```

```r
Chroms <- 1:12


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)
```

```r
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 78863 SNPs

Filtering by total sample read depth: Total DP >= 100

...Filtered 415979 SNPs

Filtering by total sample read depth: Total DP <= 400

...Filtered 4576 SNPs

Filtering by per sample read depth: DP >= 40

...Filtered 5778 SNPs

Original SNP number: 1304487, Filtered: 505196, Remaining: 799291
```

```r
#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...
```

```r
#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: 40-197

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

Simulating 10000 SNPs with reads at each depth: 40-197

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

Joining, by = "tricubeDP"
```

```r
#Plot G Statistic Distribution
hist(df_filt2$G,breaks = 950,xlim = c(0,10),xlab = "G Distribution",main = "Histogram of G Values")
```

```
# G' Distribution Plot
plotGprimeDist_MH(SNPset = df_filt2, outlierFilter = "Hampel")
```

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

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

```
plotGprimeDist_MH(SNPset = df_filt2, outlierFilter = "deltaSNP",filterThreshold = 0.1)
```

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

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

```
#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
```

```
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
```

```
deltaSNP<-plotQTLStats(SNPset = df_filt2, var = "deltaSNP", plotIntervals  = TRUE)
ggsave(filename = "DeltaSNPInterval.png",plot = last_plot())
```

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

```
neglog<-plotQTLStats(SNPset = df_filt2, var = "negLog10Pval",plotThreshold = TRUE,q=0.01)
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("8"))
#plot the plots
snpnumber
```

1
2
3
4
5
6
7
8
9
10
11
12
Number of SNPs in window
40000
30000
20000
10000
0
Genomic Position (Mb)

```
Gprime
```

```
Gprime2
```

```
deltaSNP
```

1
2
3
4
5
6
7
8
9
10
11
12
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

QTLTable <- getQTLTable(SNPset = df_filt2, 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)



#Use the function to plot allele frequencies per chromosome
Obs_Allele_Freq(SNPSet = df_filt)
```

Low Bulk Observed High Parent Allele Frequency
Allele Frequency
1.00
0.75
0.50
0.25
0.00
0e+00
2e+05
4e+05
6e+05
8e+05
SNP
factor(CHROM)
1
2
3
4
5
6
7
8
9
10
11
12

```
##Use the function to investigate chromosomal region of interest
Obs_Allele_Freq2(SNPSet = df_filt, ChromosomeValue = 8, threshold = .90)
```

Low Bulk Observed High Parent Allele Frequency
17241691
24525659
0.543
0.540
0.537
0.534
Allele Frequency
1.00
1.25
1.50
1.75
2.00
SNP
factor(CHROM)
8

```
[1] CHROM            POS              p1               p2
[5] AD_High          AD_Low           Gprime           SNP_Observations
<0 rows> (or 0-length row.names)
```