

QTL-Sorghum

Michael Hall

3/2/2022

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

Downloading GitHub repo PBGLMichaelHall/QTLseqr@HEAD

```
* checking for file '/tmp/Rtmp00kH7r/remotes2f256e86d541/PBGLMichaelHall-QTLseqr-856f201/DESCRIPTION' .
* preparing 'QTLseqr':
* checking DESCRIPTION meta-information ... OK
* cleaning src
Warning: /tmp/Rtmp00kH7r/Rbuild2f67649838af/QTLseqr/man/tricube_Smooth.Rd:2: unexpected '}'
Warning: /tmp/Rtmp00kH7r/Rbuild2f67649838af/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)
```

```
#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 variant 5000

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 ...
$ 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 ...
$ 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,
```

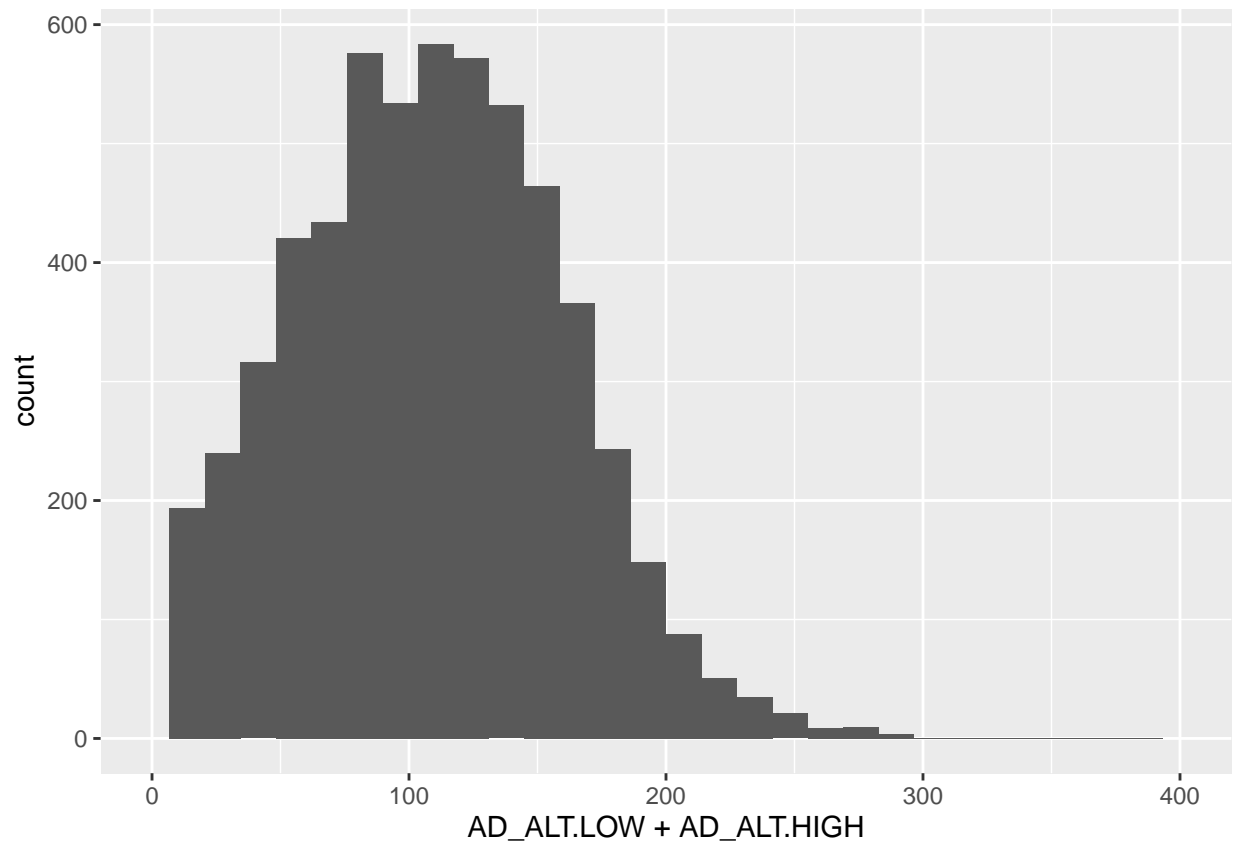
```
#the tidy data frame makes a CHROMKEY so no need to change chromosome names
```

```
Chroms <- 1:10
```

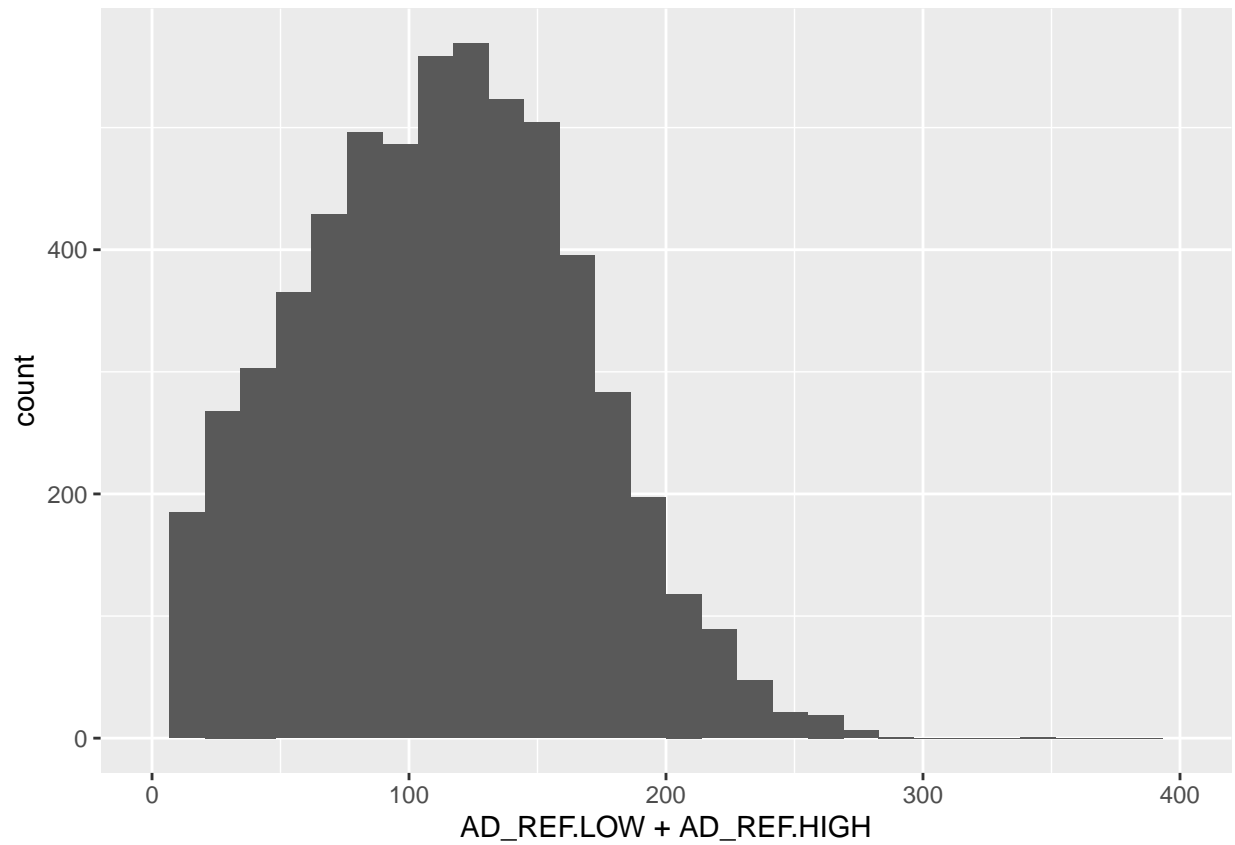
```
df <-
  importFromTable(
    file = file,
    highBulk = HighBulk,
    lowBulk = LowBulk,
    chromList = Chroms
  )

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

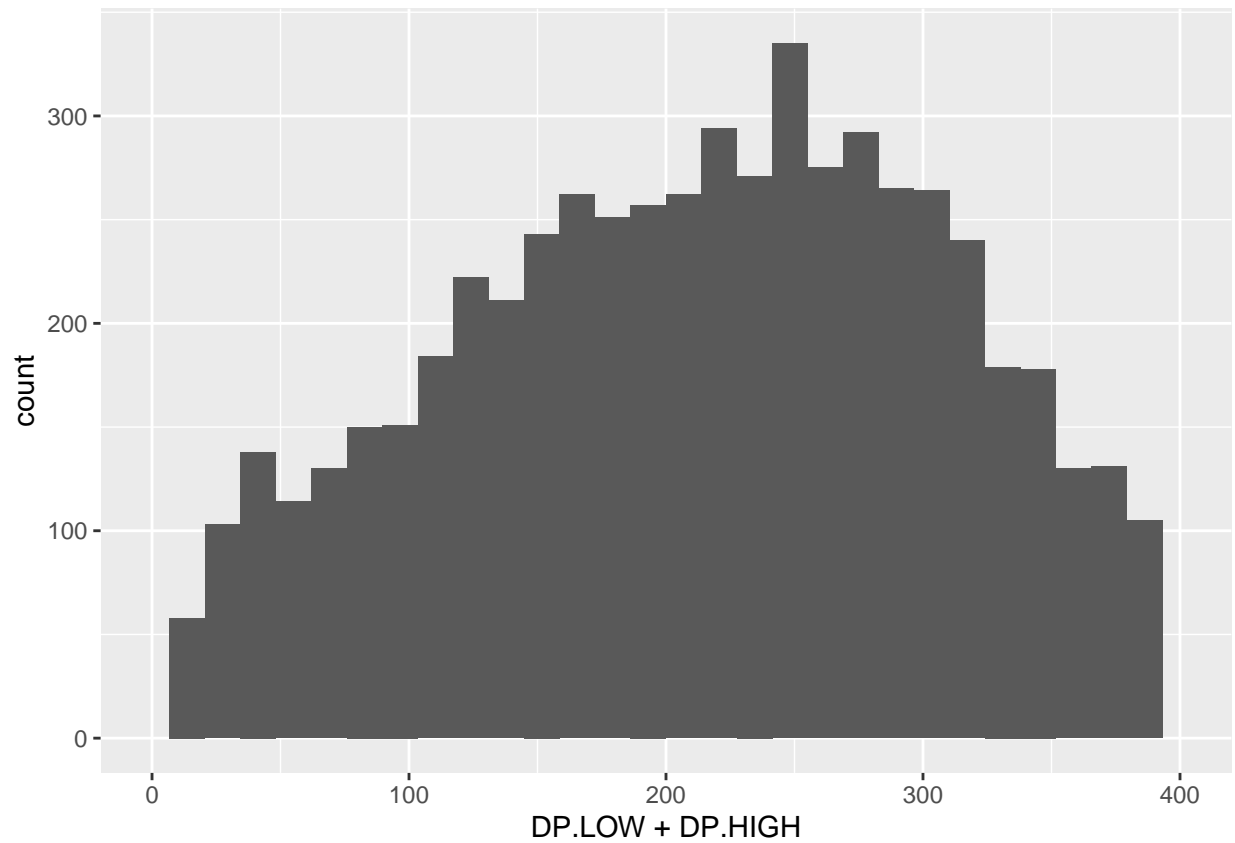
ggplot(data = df) +
  geom_histogram(aes(x = AD_ALT.LOW + AD_ALT.HIGH)) + xlim(0,400)
```



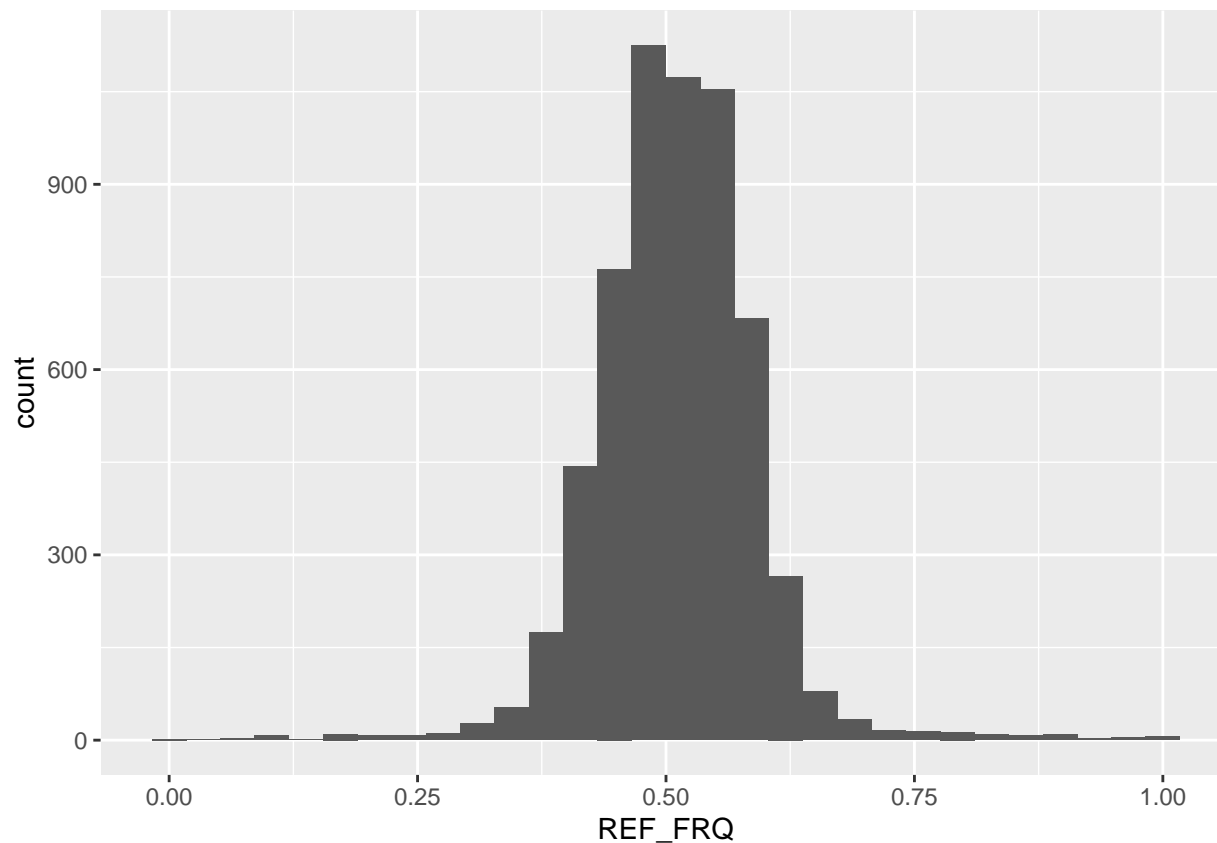
```
ggsave(filename = "AD_Histogram.png", plot = last_plot())
ggplot(data = df) +
  geom_histogram(aes(x = AD_REF.LOW + AD_REF.HIGH)) + xlim(0,400)
```



```
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...

```
#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

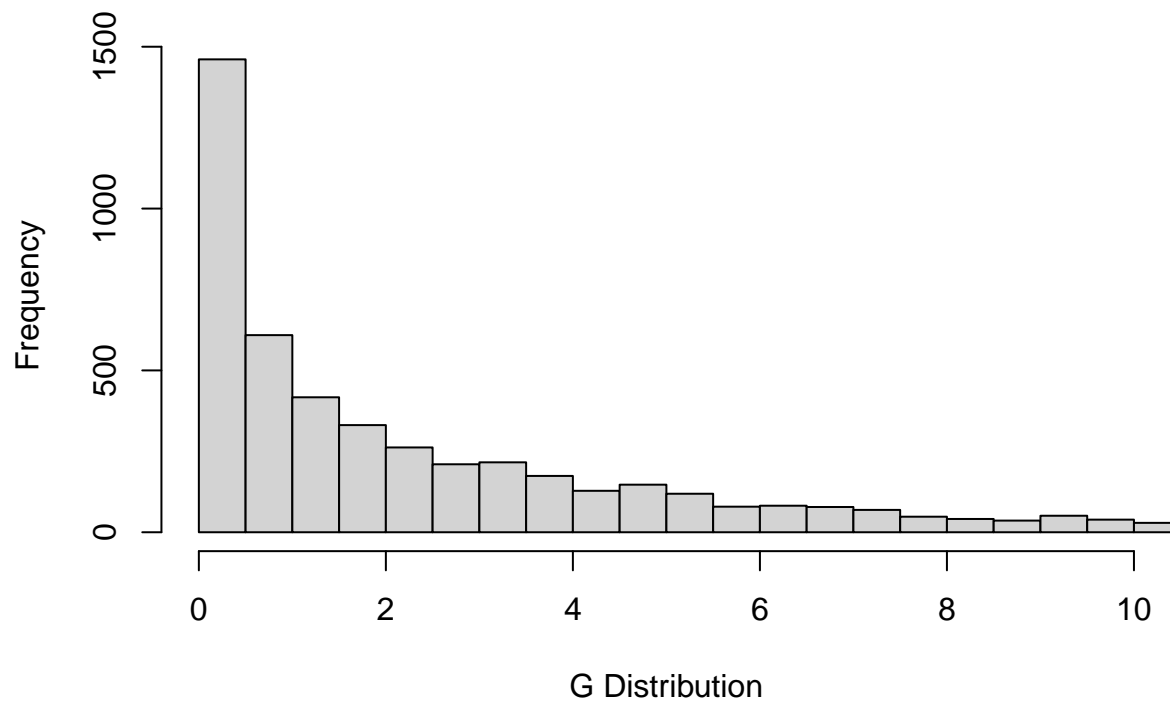
Joining, by = "tricubeDP"

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

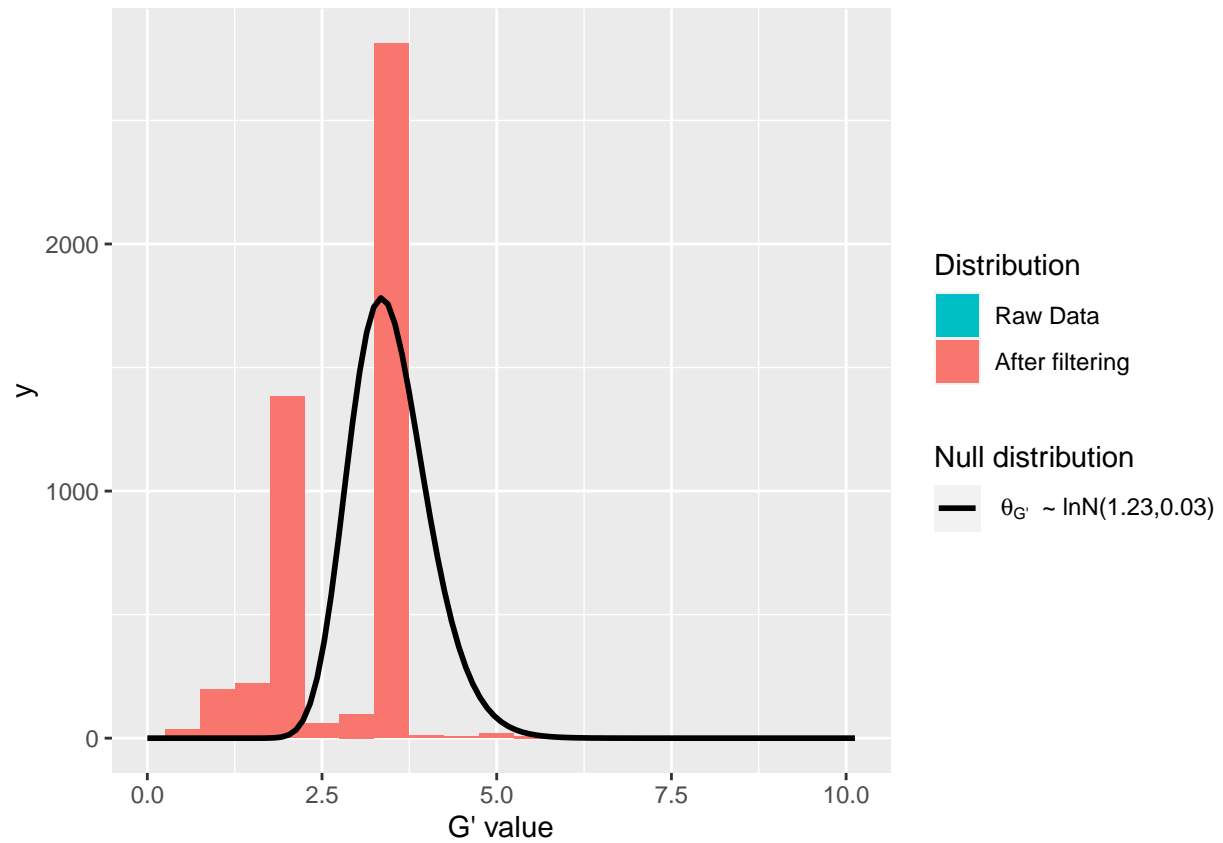
```
#Plot G Statistic Distribution
```

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

Histogram of G Values



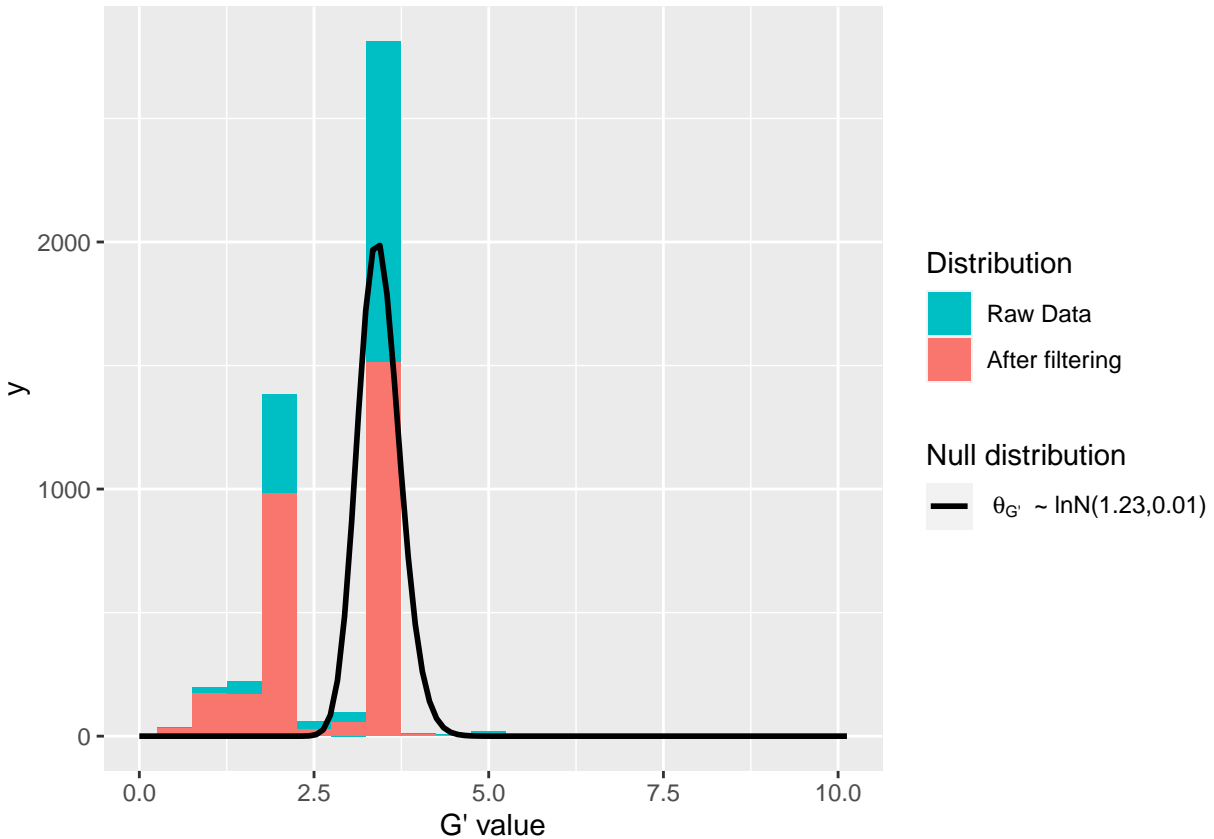
```
# G' Distribution Plot  
plotGprimeDist_MH(SNPset = df_filt2, outlierFilter = "Hampel")
```

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

Saving 6.5 x 4.5 in image

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

```
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

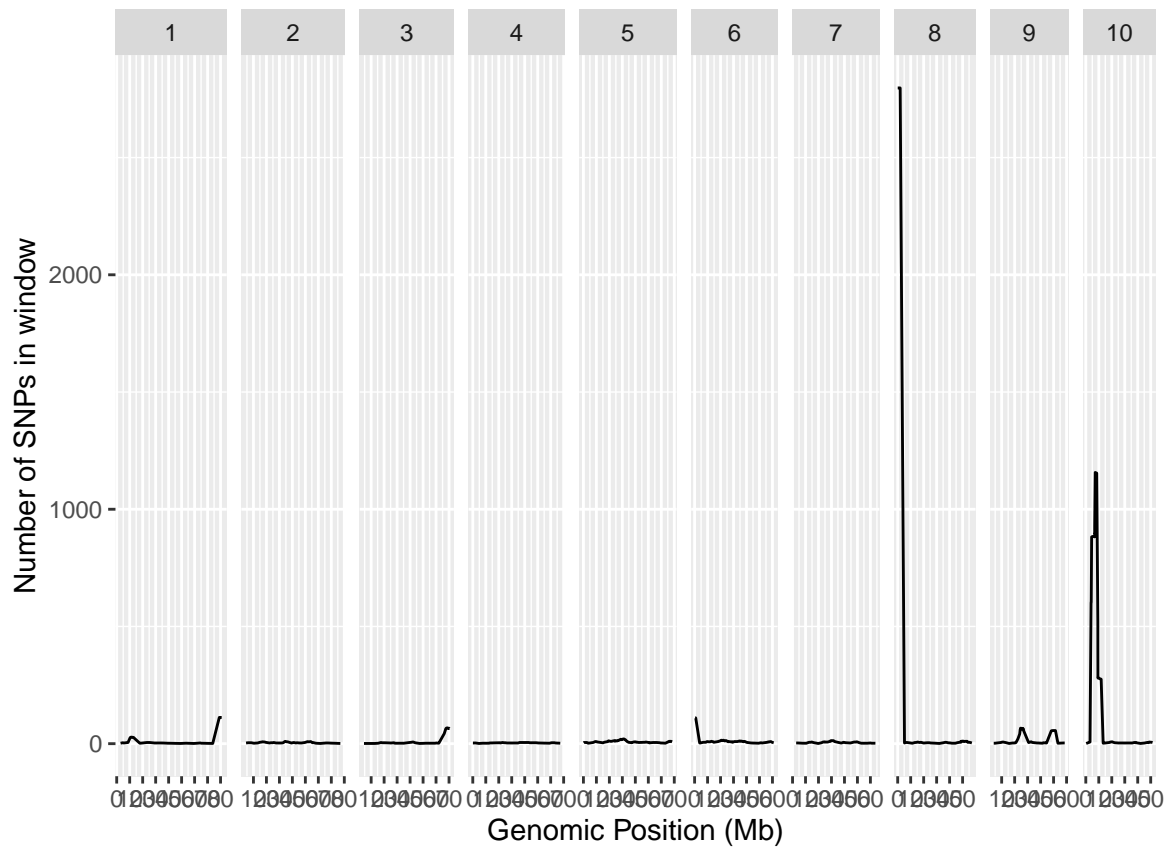
```
setwd("/home/michael/Desktop/SorghumQTL/negLog10Pval/")
```

```
neglog <- plotQTLStats(SNPset = df_filt2, var = "negLog10Pval", plotThreshold = TRUE, q = 0.01, subset = c("1"))
```

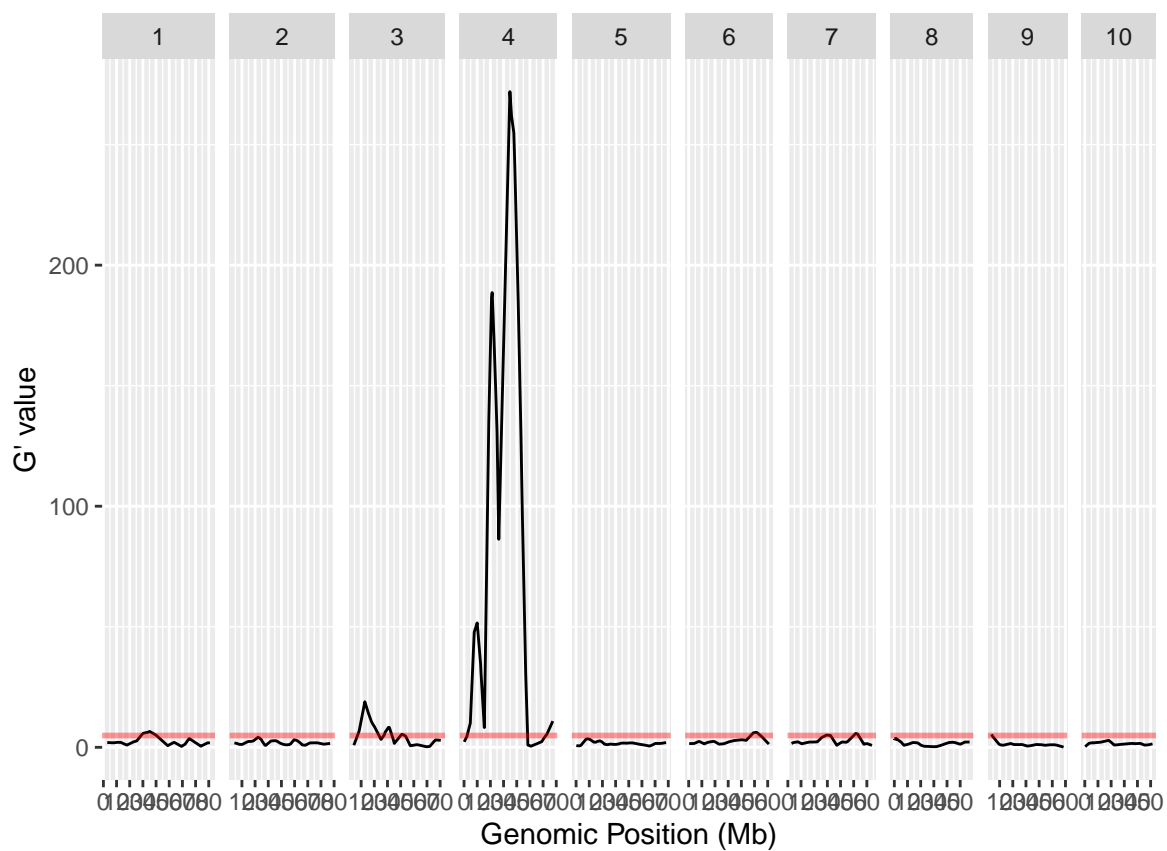
```
ggsave(filename = "negLog10Pval.png", plot = last_plot())
```

Saving 6.5 x 4.5 in image

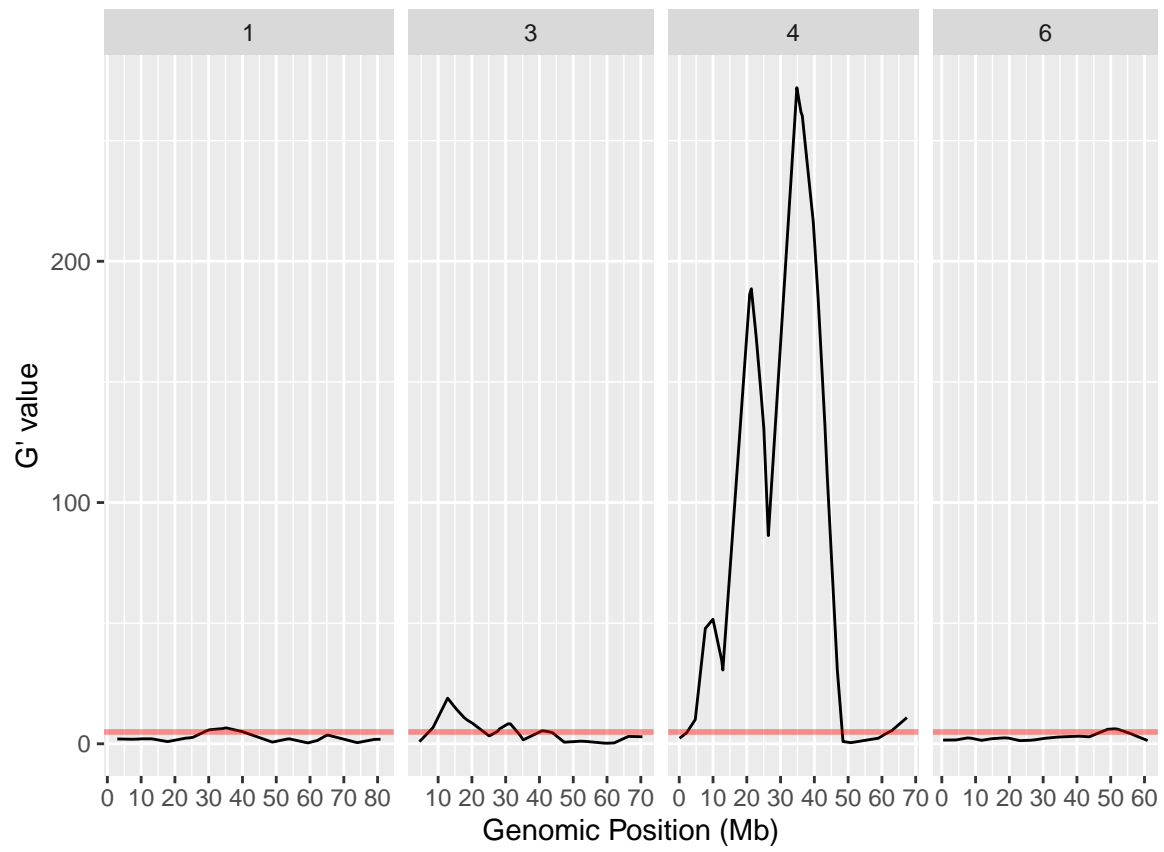
```
Gprime2<-plotQTLStats(SNPset = df_filt2, var = "Gprime",plotThreshold = TRUE,q=0.01,subset = c("1","3",
#plot the plots
snpnumber
```



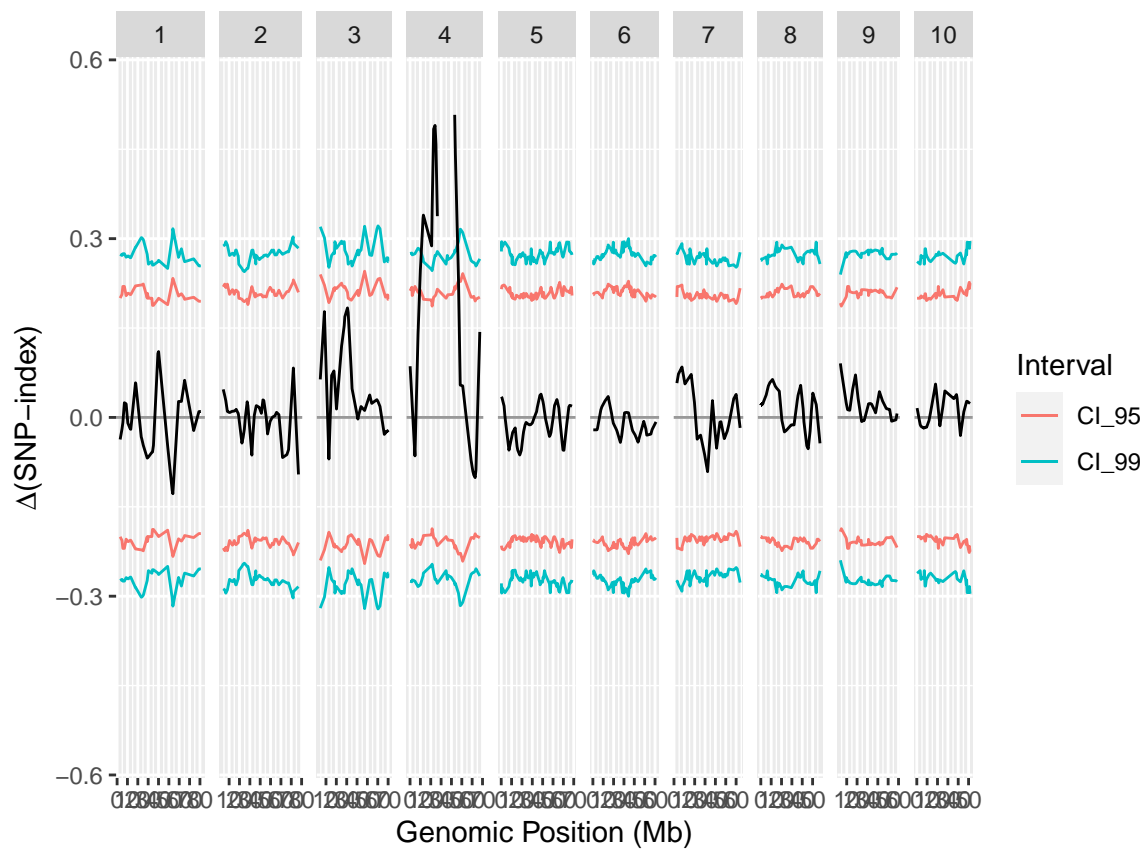
Gprime



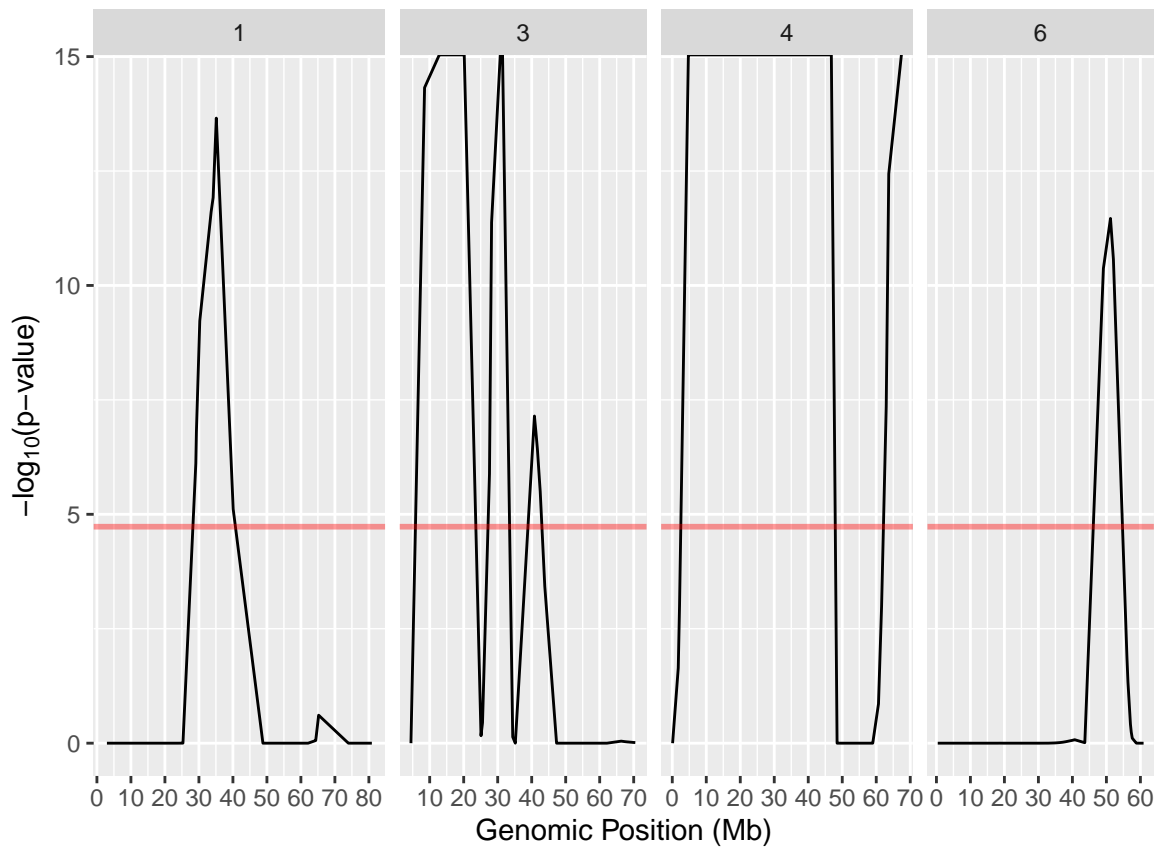
Gprime2



deltaSNP

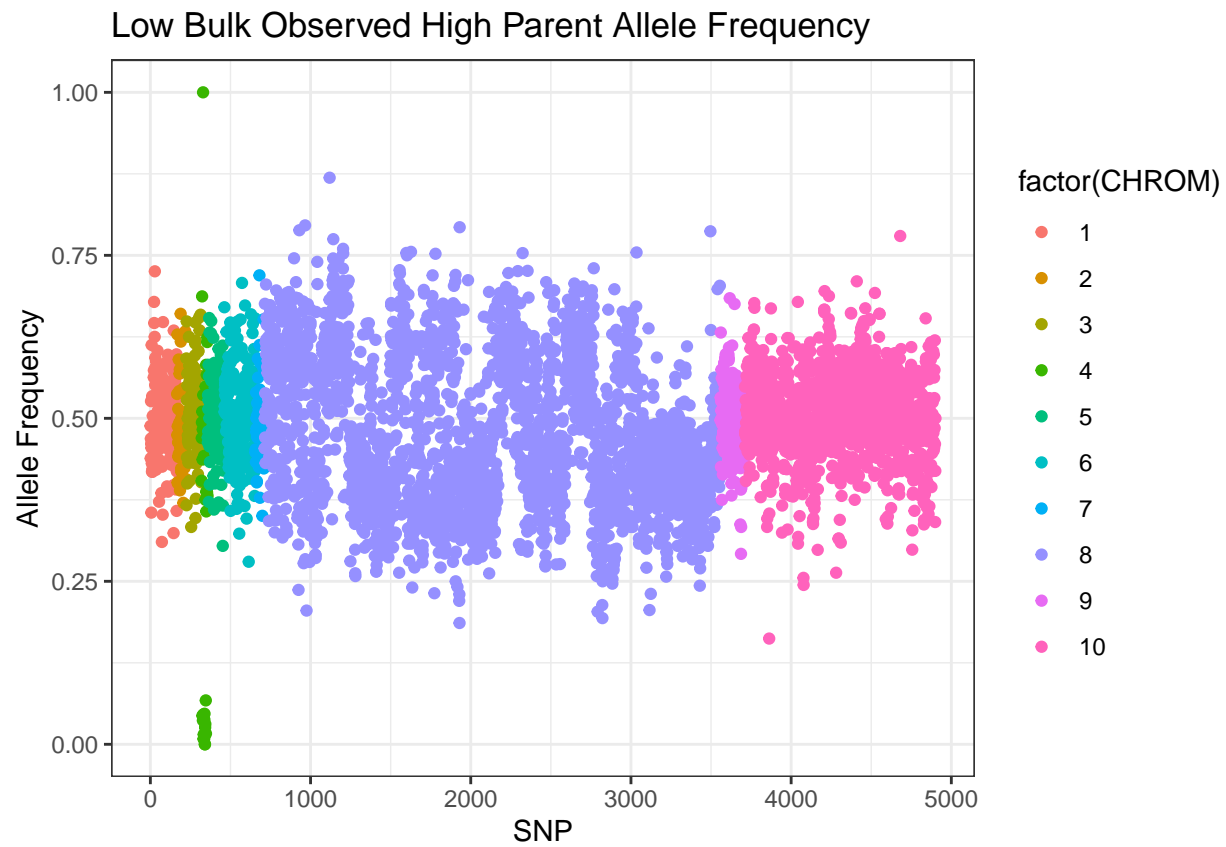


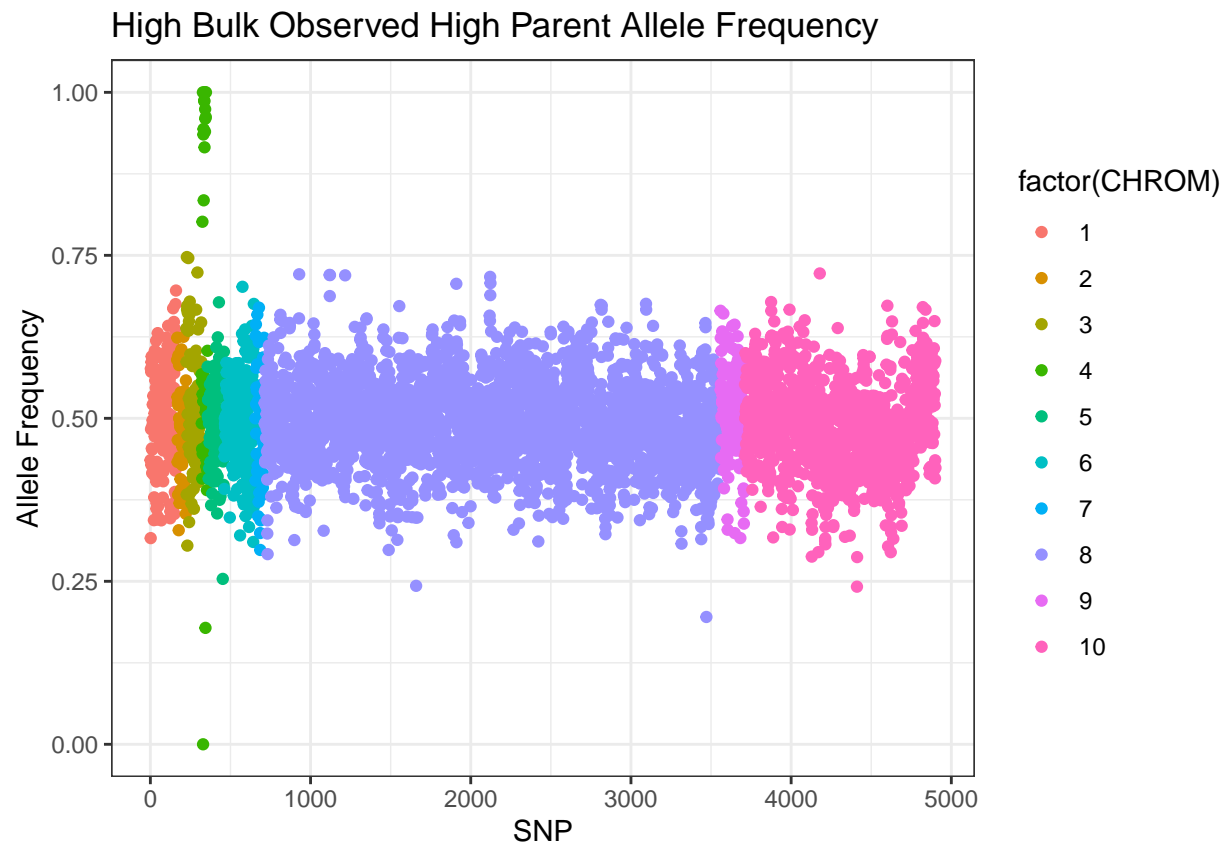
neglog



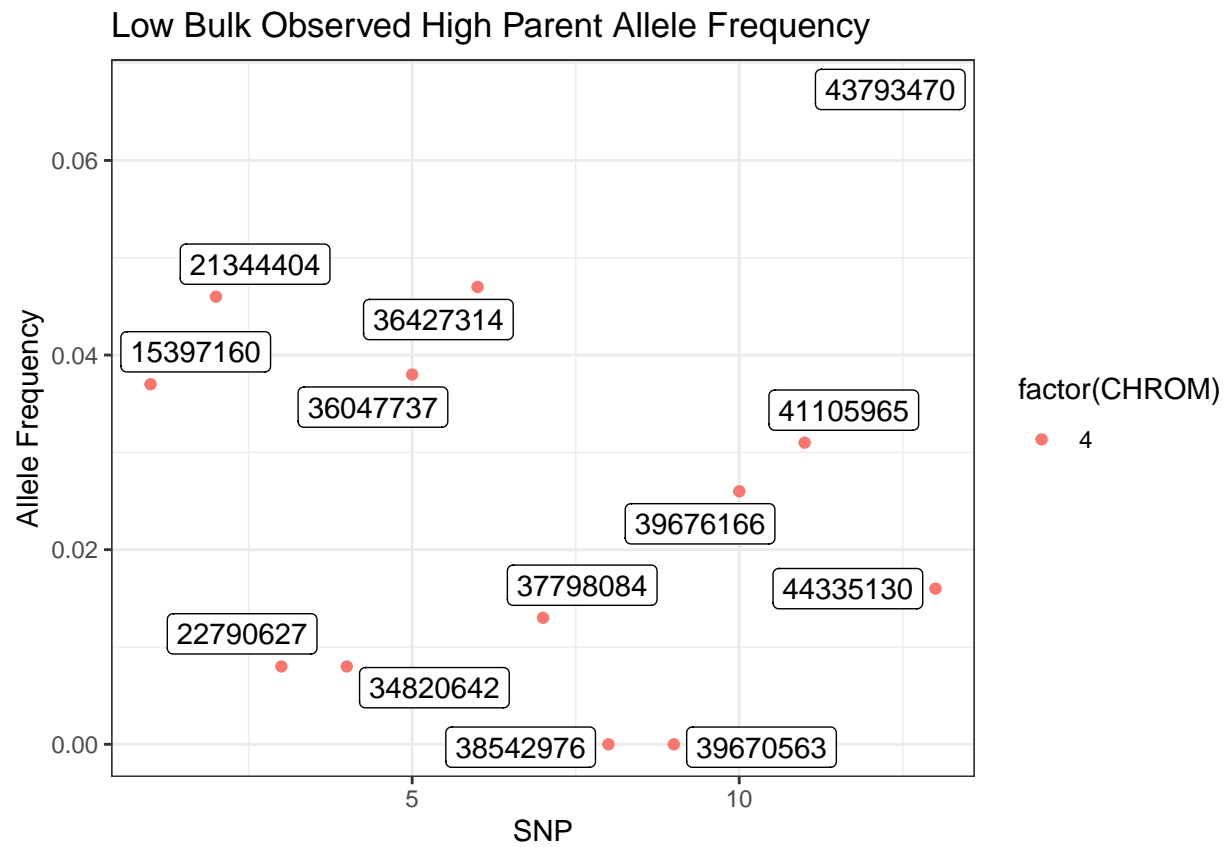
```
#export summary CSV
setwd("/home/michael/Desktop/SorghumQTL/PeakSummary/")
QTLTable <- getQTLTable(SNPset = df_filt, alpha = 0.01, export = TRUE, fileName = "my_BSA_QTL.csv")
write.csv(QTLTable, file = "QTLTablePeaks.csv", row.names = FALSE, col.names = TRUE)
Table4 <- read.table(file = "QTLTablePeaks.csv",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

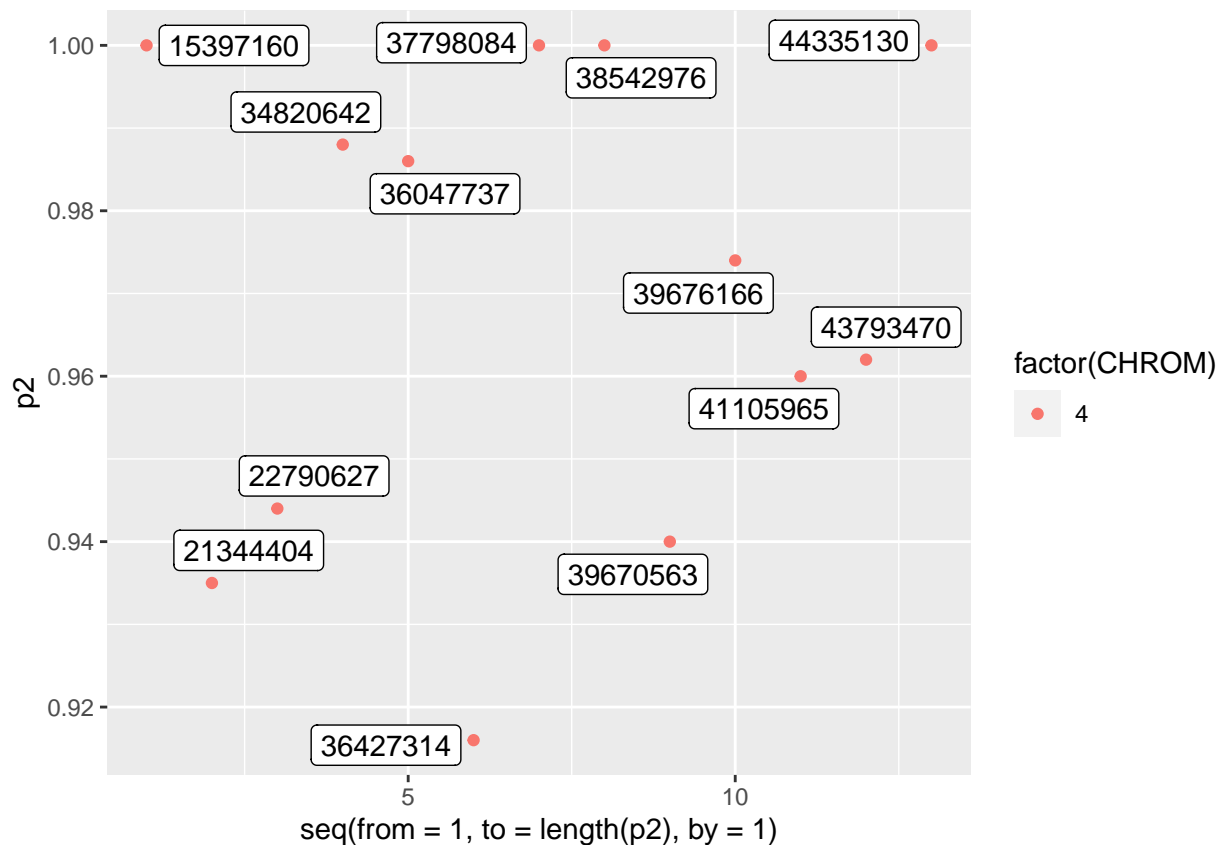
#Use the function to plot allele frequencies per chromosome
Obs_Allele_Freq(SNPSet = df_filt)
```





```
##Use the function to investigate chromosomal region of interest  
Obs_Allele_Freq2(SNPSet = df_filt, ChromosomeValue = 4, threshold = .90)
```

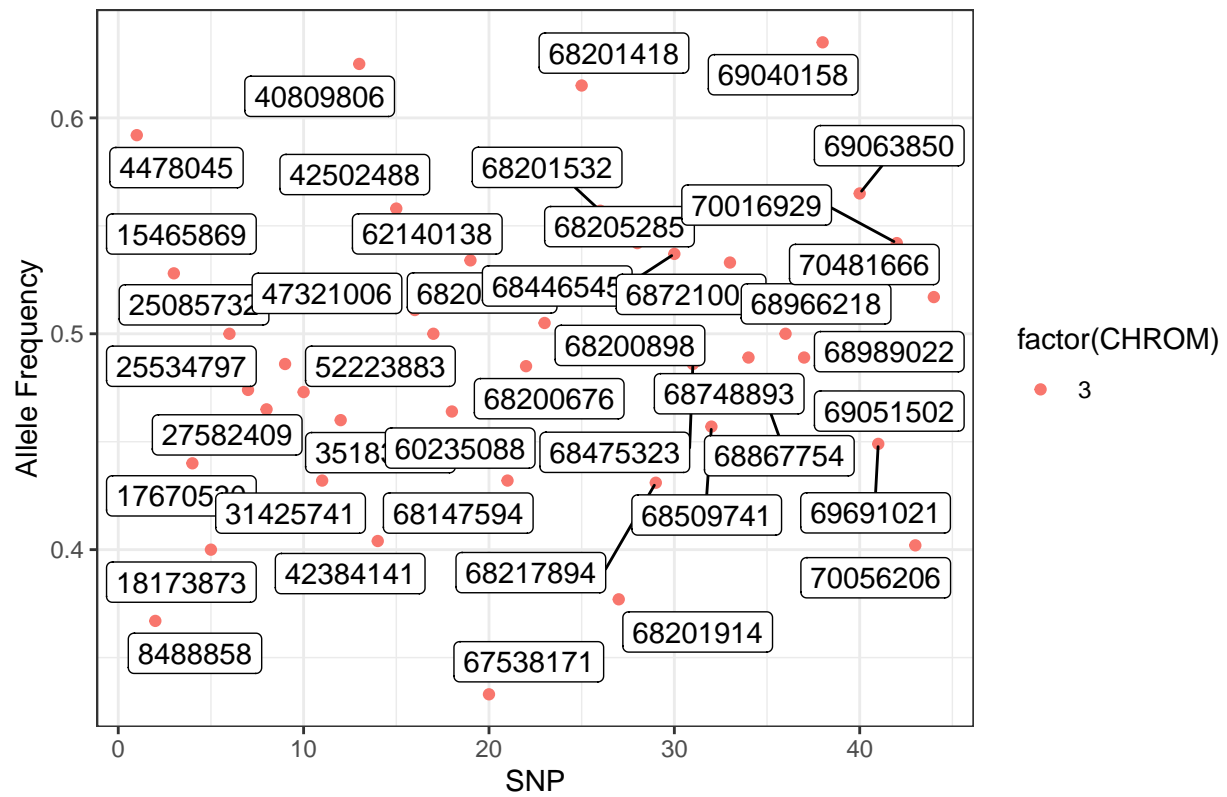


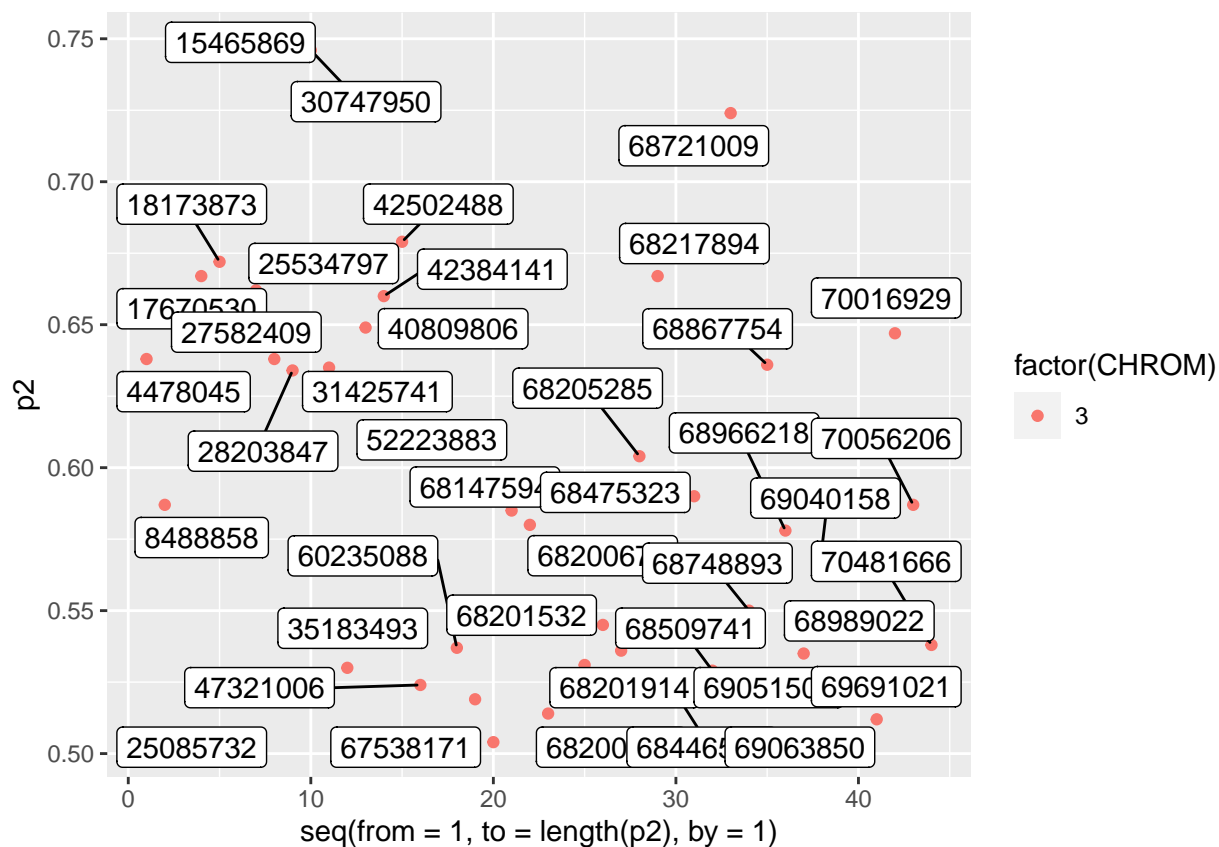


	CHROM	POS	p1	p2	AD_High	AD_Low	Gprime
328	4	15397160	0.037	1.000	76,3	0,79	8.192
331	4	21344404	0.046	0.935	159,6	11,125	188.612
332	4	22790627	0.008	0.944	151,1	9,120	168.546
336	4	34820642	0.008	0.988	159,1	2,117	271.960
337	4	36047737	0.038	0.986	143,5	2,127	261.796
338	4	36427314	0.047	0.916	76,3	7,61	260.301
339	4	37798084	0.013	1.000	152,2	0,149	254.904
340	4	38542976	0.000	1.000	89,0	0,59	241.256
341	4	39670563	0.000	0.940	78,0	5,53	216.461
342	4	39676166	0.026	0.974	113,2	3,75	216.337
343	4	41105965	0.031	0.960	190,5	8,156	184.998
346	4	43793470	0.067	0.962	127,7	5,97	111.475
347	4	44335130	0.016	1.000	66,1	0,60	94.982

```
Obs_Allele_Freq2(SNPSet = df_filt, ChromosomeValue = 3, threshold = .50)
```

Low Bulk Observed High Parent Allele Frequency

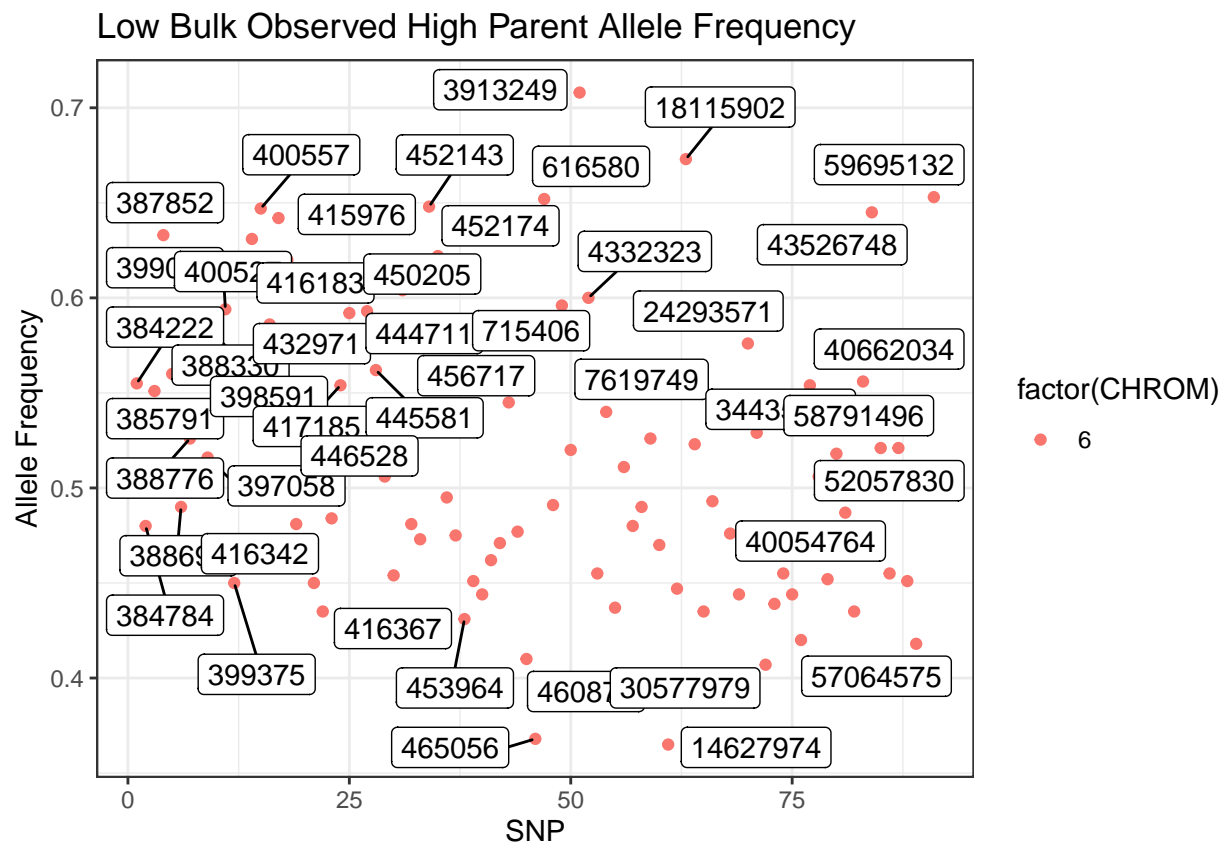


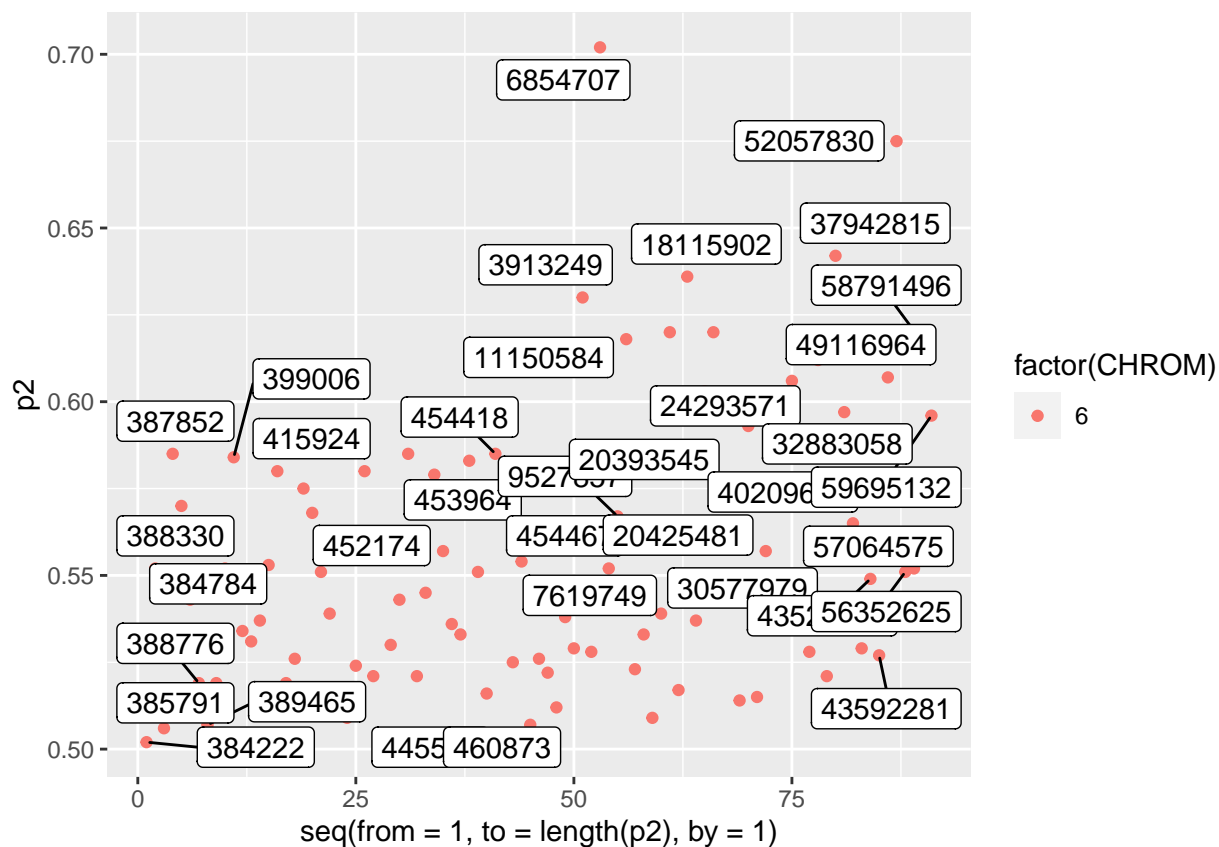


	CHROM	POS	p1	p2	AD_High	AD_Low	Gprime
225	3	4478045	0.592	0.638	37,29	21,20	0.877
226	3	8488858	0.367	0.587	44,22	31,38	6.708
228	3	15465869	0.528	0.747	142,66	48,59	14.266
229	3	17670530	0.440	0.667	38,22	19,28	10.890
230	3	18173873	0.400	0.672	78,34	38,51	10.332
233	3	25085732	0.500	0.506	40,36	39,36	3.282
234	3	25534797	0.474	0.662	43,27	22,30	3.536
235	3	27582409	0.465	0.638	67,33	38,38	5.150
236	3	28203847	0.486	0.634	85,52	49,55	6.197
237	3	30747950	0.473	0.746	44,35	15,39	8.295
238	3	31425741	0.432	0.635	139,54	80,71	8.241
240	3	35183493	0.460	0.530	35,29	31,34	1.615
241	3	40809806	0.625	0.649	61,50	33,30	5.408
245	3	42384141	0.404	0.660	64,40	33,59	5.097
246	3	42502488	0.558	0.679	91,43	43,34	5.070
248	3	47321006	0.511	0.524	33,24	30,23	0.656
249	3	52223883	0.500	0.598	64,35	43,35	1.085
252	3	60235088	0.464	0.537	36,26	31,30	0.256
253	3	62140138	0.534	0.519	41,31	38,27	0.378
255	3	67538171	0.333	0.504	62,38	61,76	3.018
259	3	68147594	0.432	0.585	79,35	56,46	2.998
262	3	68200676	0.485	0.580	76,48	55,51	2.996
263	3	68200838	0.505	0.514	55,49	52,48	2.996
264	3	68200898	0.496	0.507	73,58	71,59	2.996
279	3	68201418	0.615	0.531	104,99	92,62	2.996

280	3	68201532	0.557	0.545	104,102	87,81	2.996
284	3	68201914	0.377	0.536	98,58	85,96	2.996
285	3	68205285	0.542	0.604	32,26	21,22	2.996
286	3	68217894	0.431	0.667	62,31	31,41	2.995
289	3	68446545	0.537	0.521	50,43	46,37	2.988
290	3	68475323	0.486	0.590	105,70	73,74	2.986
292	3	68509741	0.457	0.529	55,32	49,38	2.985
294	3	68721009	0.533	0.724	55,32	21,28	2.977
296	3	68748893	0.489	0.550	71,44	58,46	2.976
298	3	68867754	0.468	0.636	56,37	32,42	2.971
301	3	68966218	0.500	0.578	48,32	35,32	2.968
302	3	68989022	0.489	0.535	92,64	80,67	2.967
307	3	69040158	0.635	0.572	83,80	62,46	2.965
308	3	69051502	0.462	0.527	68,48	61,56	2.964
310	3	69063850	0.565	0.507	70,78	68,60	2.964
314	3	69691021	0.449	0.512	103,75	98,92	2.940
317	3	70016929	0.542	0.647	44,26	24,22	2.928
318	3	70056206	0.402	0.587	71,43	50,64	2.926
319	3	70481666	0.517	0.538	85,78	73,73	2.910

```
Obs_Allele_Freq2(SNPSet = df_filt, ChromosomeValue = 6, threshold = .50)
```



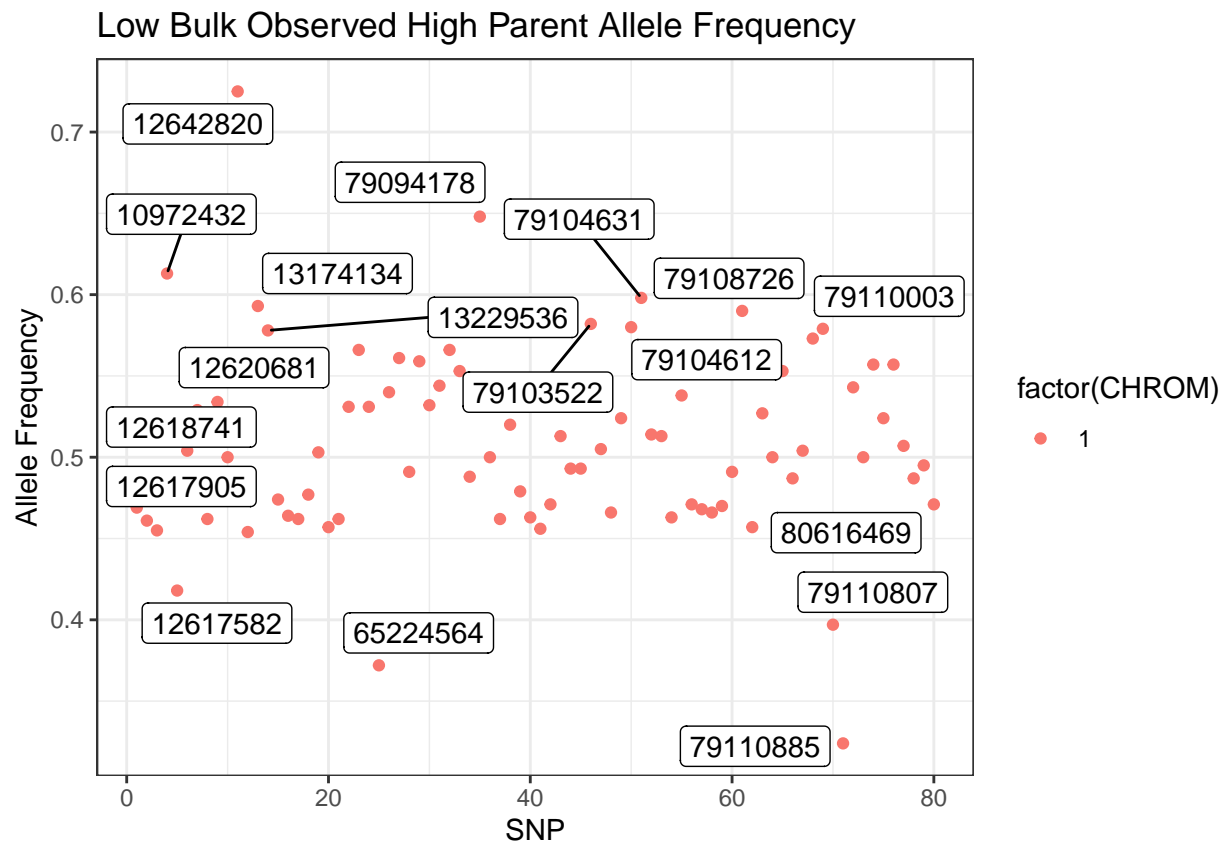


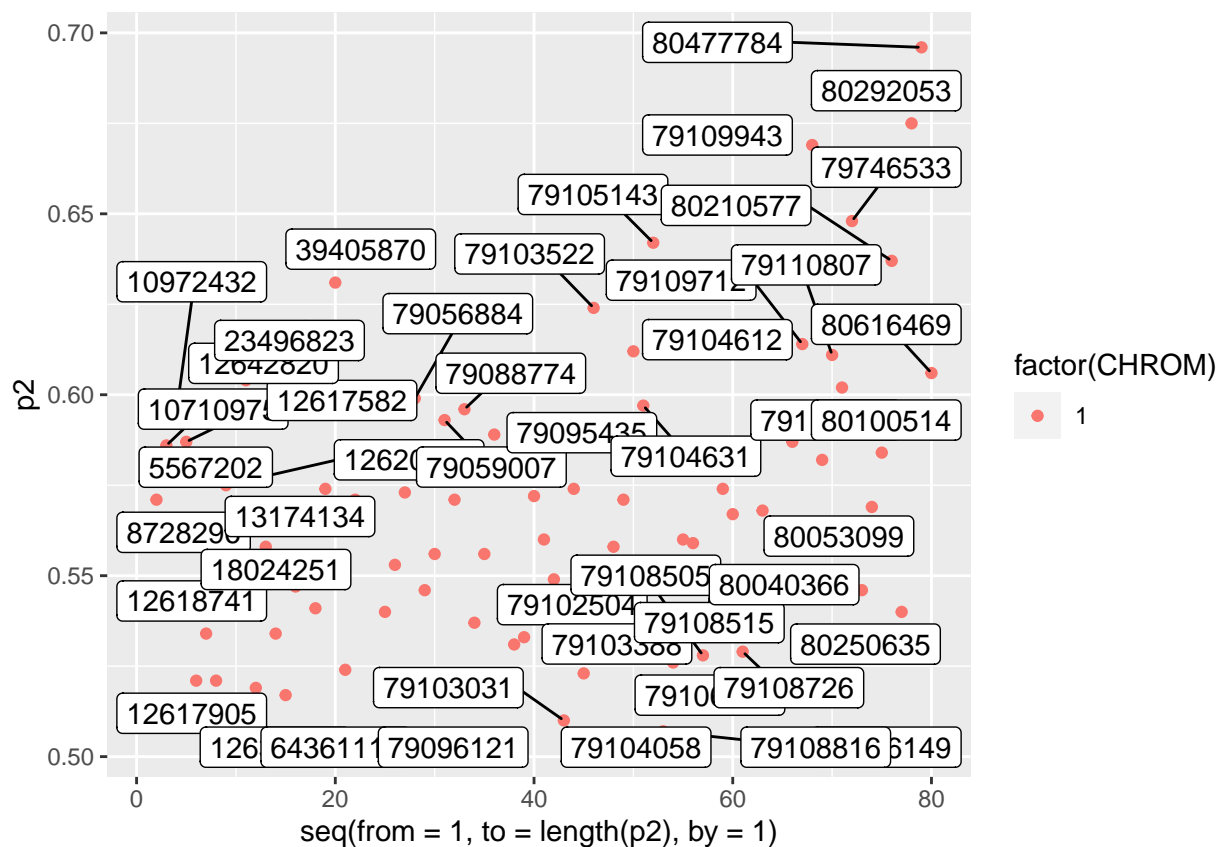
	CHROM	POS	p1	p2	AD_High	AD_Low	Gprime
469	6	384222	0.555	0.502	103,76	102,61	1.539
473	6	384784	0.480	0.552	96,73	78,79	1.539
474	6	385791	0.551	0.506	85,81	83,66	1.539
475	6	387852	0.633	0.585	103,105	73,61	1.539
476	6	388330	0.560	0.570	69,51	52,40	1.539
477	6	388697	0.490	0.543	107,77	90,80	1.539
478	6	388776	0.526	0.519	56,40	52,36	1.539
480	6	389465	0.506	0.507	69,43	67,42	1.539
483	6	397058	0.516	0.519	69,65	64,61	1.539
487	6	398591	0.580	0.552	48,29	39,21	1.539
488	6	399006	0.594	0.584	52,41	37,28	1.539
493	6	399375	0.450	0.534	62,45	54,55	1.539
495	6	399740	0.494	0.531	76,84	67,86	1.539
498	6	400527	0.631	0.537	36,41	31,24	1.539
499	6	400557	0.647	0.553	42,44	34,24	1.539
502	6	415924	0.586	0.580	51,41	37,29	1.540
503	6	415976	0.642	0.519	41,43	38,24	1.540
504	6	416183	0.620	0.526	71,44	64,27	1.540
505	6	416252	0.481	0.575	61,26	45,28	1.540
506	6	416291	0.471	0.568	63,24	48,27	1.540
507	6	416342	0.450	0.551	59,27	48,33	1.540
508	6	416367	0.435	0.539	55,27	47,35	1.540
509	6	416386	0.484	0.510	51,30	49,32	1.540
510	6	417185	0.554	0.509	59,62	57,50	1.540
515	6	432057	0.592	0.524	66,74	60,51	1.540

516	6	432971	0.569	0.580	80,74	58,56	1.540
525	6	444711	0.593	0.521	73,64	67,44	1.540
527	6	445581	0.562	0.504	69,68	68,53	1.540
532	6	446528	0.506	0.530	97,82	86,80	1.540
534	6	448103	0.454	0.543	95,69	80,83	1.540
536	6	450205	0.604	0.585	76,55	54,36	1.540
537	6	450665	0.481	0.521	61,39	56,42	1.540
538	6	450674	0.473	0.545	66,43	55,48	1.540
543	6	452143	0.648	0.579	55,59	40,32	1.540
544	6	452174	0.622	0.557	49,61	39,37	1.540
546	6	453167	0.495	0.536	45,51	39,52	1.540
547	6	453219	0.475	0.533	56,48	49,53	1.540
548	6	453964	0.431	0.583	67,44	48,58	1.540
549	6	454157	0.451	0.551	76,46	62,56	1.540
550	6	454370	0.444	0.516	66,36	62,45	1.540
551	6	454418	0.462	0.585	69,30	49,35	1.540
552	6	454467	0.471	0.569	62,24	47,27	1.540
553	6	456717	0.545	0.525	106,91	96,76	1.540
554	6	459132	0.477	0.554	46,31	37,34	1.540
559	6	460873	0.410	0.507	77,66	75,95	1.540
561	6	465056	0.368	0.526	50,28	45,48	1.540
563	6	616580	0.652	0.522	60,58	55,31	1.542
566	6	639623	0.491	0.512	107,79	102,82	1.542
567	6	715406	0.596	0.538	71,68	61,46	1.543
568	6	720164	0.520	0.529	72,66	64,61	1.543
572	6	3913249	0.708	0.630	51,46	30,19	1.581
573	6	4332323	0.600	0.528	28,33	25,22	1.624
575	6	6854707	0.455	0.702	40,25	17,30	2.239
576	6	7619749	0.540	0.552	116,87	94,74	2.426
580	6	9527857	0.437	0.567	93,52	71,67	2.061
581	6	11150584	0.511	0.618	68,48	42,46	1.601
582	6	11731523	0.480	0.523	46,36	42,39	1.436
584	6	12004221	0.490	0.533	32,24	28,25	1.493
586	6	13011995	0.526	0.509	55,40	53,36	1.714
587	6	14373081	0.470	0.539	76,62	65,70	2.011
588	6	14627974	0.365	0.620	49,19	30,33	2.067
589	6	14735537	0.447	0.517	31,21	29,26	2.091
591	6	18115902	0.673	0.636	49,35	28,17	2.426
592	6	18118536	0.523	0.537	36,34	31,31	2.426
593	6	18399688	0.435	0.603	41,30	27,39	2.444
595	6	18675107	0.493	0.620	93,67	57,69	2.461
598	6	20393545	0.542	0.574	105,90	78,76	2.155
599	6	20425481	0.476	0.571	36,30	27,33	2.145
602	6	22043951	0.444	0.514	76,55	72,69	1.632
609	6	24293571	0.576	0.593	73,72	50,53	1.363
612	6	25872403	0.529	0.515	35,27	33,24	1.445
620	6	30577979	0.407	0.557	39,22	31,32	2.279
621	6	32392854	0.439	0.536	37,18	32,23	2.525
623	6	32883058	0.455	0.593	54,20	37,24	2.589
624	6	32883065	0.444	0.606	57,20	37,25	2.589
625	6	34368555	0.420	0.619	39,21	24,29	2.785
626	6	34435846	0.554	0.528	47,41	42,33	2.793
627	6	34770268	0.506	0.612	134,86	85,84	2.815
629	6	35637255	0.452	0.521	38,28	35,34	2.868

634	6	37942815	0.518	0.642	138,87	77,81	3.011
636	6	40054764	0.487	0.597	83,55	56,58	3.113
637	6	40209616	0.435	0.565	104,54	80,70	3.121
639	6	40662034	0.556	0.529	63,60	56,48	3.142
640	6	43526748	0.645	0.549	79,69	65,38	2.904
641	6	43592281	0.521	0.527	79,50	71,46	2.891
643	6	49116964	0.455	0.607	85,66	55,79	6.017
646	6	52057830	0.521	0.675	102,75	49,69	6.053
649	6	56352625	0.451	0.551	70,46	57,56	3.956
650	6	57064575	0.418	0.552	100,59	81,82	3.514
653	6	58791496	0.539	0.617	113,83	70,71	2.537
654	6	59695132	0.653	0.596	53,49	36,26	2.029

```
Obs_Allele_Freq2(SNPSet = df_filt, ChromosomeValue = 1, threshold = .50)
```





	CHROM	POS	p1	p2	AD_High	AD_Low	Gprime
3	1	5567202	0.469	0.576	53,45	39,51	1.929
5	1	8728296	0.461	0.571	88,59	66,69	1.929
7	1	10710975	0.455	0.586	34,20	24,24	2.053
8	1	10972432	0.613	0.595	47,49	32,31	2.052
13	1	12617582	0.418	0.587	84,56	59,78	2.051
15	1	12617905	0.504	0.521	74,67	68,66	2.051
18	1	12618741	0.529	0.534	70,55	61,49	2.051
19	1	12618821	0.462	0.521	73,61	67,71	2.051
25	1	12620681	0.534	0.575	42,31	31,27	2.051
26	1	12638456	0.500	0.508	66,54	64,54	2.051
28	1	12642820	0.725	0.604	32,37	21,14	2.051
29	1	12643048	0.454	0.519	42,44	39,53	2.051
31	1	13174134	0.593	0.558	101,86	80,59	2.043
32	1	13229536	0.578	0.534	55,48	48,35	2.042
33	1	17781204	0.474	0.517	45,27	42,30	0.915
34	1	18024251	0.464	0.547	70,45	58,52	0.961
36	1	23496823	0.462	0.619	39,24	24,28	2.372
40	1	29094283	0.477	0.541	33,21	28,23	5.198
44	1	34182892	0.503	0.574	120,93	89,92	6.293
46	1	39405870	0.457	0.631	111,59	65,70	5.235
47	1	40077801	0.462	0.524	66,43	60,50	4.985
51	1	59317706	0.531	0.571	92,60	69,53	0.373
52	1	62114296	0.566	0.564	57,56	44,43	1.335
53	1	64361116	0.531	0.504	64,52	63,46	3.100
54	1	65224564	0.372	0.540	87,48	74,81	3.637

58	1	79054218	0.540	0.553	119,95	96,81	1.824
59	1	79055107	0.561	0.573	82,60	61,47	1.824
64	1	79056884	0.491	0.599	115,80	77,83	1.824
65	1	79057099	0.559	0.546	89,95	74,75	1.824
66	1	79057312	0.532	0.556	70,67	56,59	1.824
67	1	79059007	0.544	0.593	54,62	37,52	1.824
70	1	79088562	0.566	0.571	101,86	76,66	1.825
71	1	79088774	0.553	0.596	106,68	72,55	1.825
75	1	79092267	0.488	0.537	73,62	63,65	1.825
79	1	79094178	0.648	0.556	85,68	68,37	1.825
83	1	79095435	0.500	0.589	89,55	62,55	1.825
85	1	79096121	0.462	0.505	103,72	101,84	1.825
86	1	79096428	0.520	0.531	76,65	67,60	1.825
92	1	79102504	0.479	0.533	96,80	84,87	1.825
93	1	79102565	0.463	0.572	87,68	65,79	1.825
94	1	79102571	0.456	0.560	84,68	66,81	1.825
95	1	79102581	0.471	0.549	79,66	65,74	1.825
96	1	79103031	0.513	0.510	50,39	48,37	1.825
99	1	79103342	0.493	0.574	54,37	40,38	1.825
100	1	79103388	0.493	0.523	58,33	53,34	1.825
101	1	79103522	0.582	0.624	78,57	47,41	1.825
103	1	79104058	0.505	0.504	67,51	66,50	1.825
105	1	79104519	0.466	0.558	53,34	42,39	1.825
106	1	79104550	0.524	0.571	56,43	42,39	1.825
108	1	79104612	0.580	0.612	52,47	33,34	1.825
109	1	79104631	0.598	0.597	46,49	31,33	1.825
111	1	79105143	0.514	0.642	52,37	29,35	1.825
115	1	79106149	0.513	0.507	69,59	67,56	1.825
116	1	79106184	0.463	0.526	71,44	64,51	1.825
118	1	79106724	0.538	0.560	84,84	66,72	1.825
120	1	79107194	0.471	0.559	104,73	82,82	1.825
127	1	79108505	0.468	0.528	95,87	85,99	1.825
128	1	79108515	0.466	0.550	82,69	67,79	1.825
129	1	79108524	0.470	0.574	74,62	55,70	1.825
130	1	79108571	0.491	0.567	72,56	55,58	1.825
132	1	79108726	0.590	0.529	74,85	66,59	1.825
133	1	79108816	0.457	0.504	62,58	61,69	1.825
135	1	79109151	0.527	0.568	109,78	83,70	1.825
136	1	79109159	0.500	0.596	90,65	61,65	1.825
137	1	79109200	0.553	0.596	56,57	38,46	1.825
139	1	79109703	0.487	0.587	81,58	57,61	1.825
140	1	79109712	0.504	0.614	86,61	54,60	1.825
142	1	79109943	0.573	0.669	107,67	53,50	1.825
143	1	79110003	0.579	0.582	96,81	69,59	1.825
144	1	79110807	0.397	0.611	69,31	44,47	1.825
145	1	79110885	0.324	0.602	62,23	41,48	1.825
147	1	79746533	0.543	0.648	149,88	81,74	1.841
148	1	80040366	0.500	0.546	100,83	83,83	1.848
149	1	80053099	0.557	0.569	111,88	84,70	1.849
151	1	80100514	0.524	0.584	59,55	42,50	1.850
154	1	80210577	0.557	0.637	114,68	65,54	1.852
155	1	80250635	0.507	0.540	88,75	75,73	1.853
157	1	80292053	0.487	0.675	79,55	38,58	1.855
160	1	80477784	0.495	0.696	87,49	38,50	1.859

162 1 80616469 0.471 0.606 66,40 43,45 1.862