qtl2_test

Bora Kim

February 7, 2022

1. Start your engine

```
library(qtl2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
## x readr::read_csv() masks qtl2::read_csv()
```

write control file and read

```
control_test <- file.path("C:/Users/Bora Kim/Desktop/Bora Kim/6. RIL_QTL/RIL_data_analysis", "control_t
write_control_file(control_test,
                   crosstype="riself",
                   geno_file="genotypes.csv",
                   gmap_file="physical_map_mb.csv", #physical map in Mb
                   pmap_file="physical_map_mb.csv",
                   pheno_file="phenotypes_median_test.csv",
                   geno_codes=c(A=1L, B=2L),
                   alleles=c("A", "B"),
                   na.strings=c("-", "NA"))
```

```
test <- read_cross2("C:/Users/Bora Kim/Desktop/Bora Kim/6. RIL_QTL/RIL_data_analysis/control_test.yaml"
```

2. QTL anlaysis

Since we don't have a genetic map that is used for probability calculation, here genetic map was estimated.

```
est<-est_map(test, cores=parallel::detectCores()-1)
test$gmap<-est #replace gmap object
```

Insert pseudomarkers into estimated genetic map

```
map <- insert_pseudomarkers(map=test$gmap, step=0.1)
```

Compares the number of markers between the estimated genetic map and the pseudomarkers-inserted map

```
sapply(test$gmap, length)
```

```
## ch01 ch02 ch03 ch04 ch05 ch06 ch07 ch08 ch09 ch10 ch11 ch12
##  508  412  438  398  349  329  327  392  447  332  304  279
```

```
sapply(map, length)
```

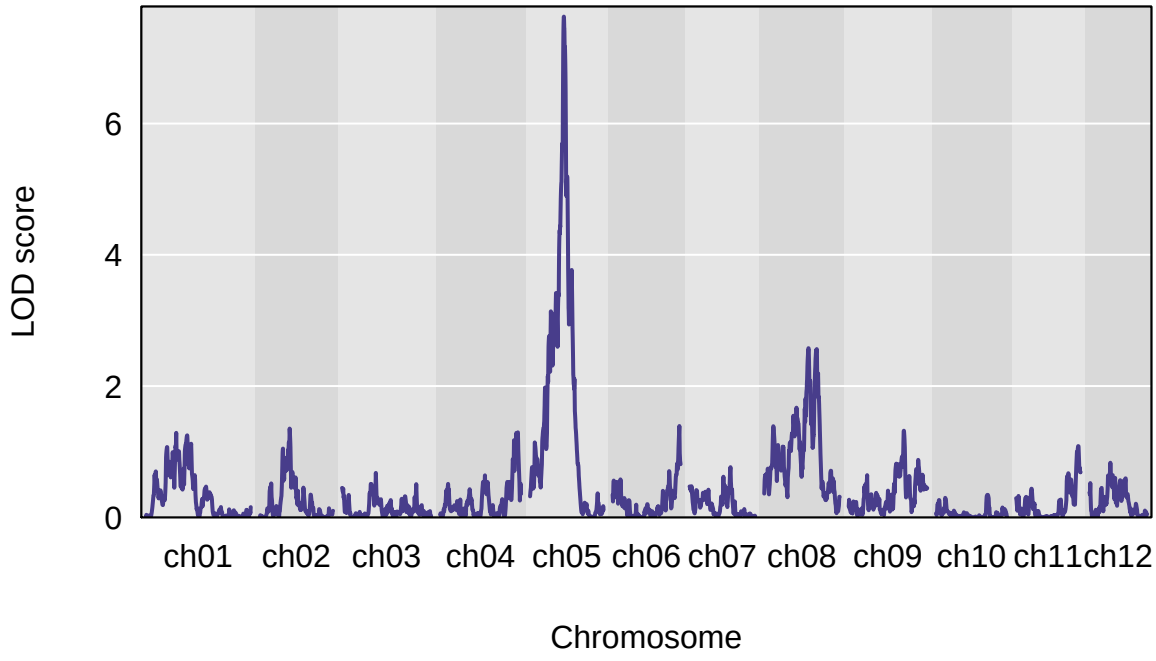
```
## ch01 ch02 ch03 ch04 ch05 ch06 ch07 ch08 ch09 ch10 ch11 ch12
## 1827 1343 1566 1417 1278 1194 1160 1348 1441 1229 1119 1003
```

Calculate genotype probability based on pseudomarkers-inserted map

```
pr <- calc_genoprob(cross=test, map=map, error_prob=0.001)
```

Finding LOD peaks

```
out <- scan1(genoprobs = pr, pheno = test$pheno, cores=parallel::detectCores()-1)
plot_scan1(out, map = map, lodcolumn = "fresh_shoot_g")
```

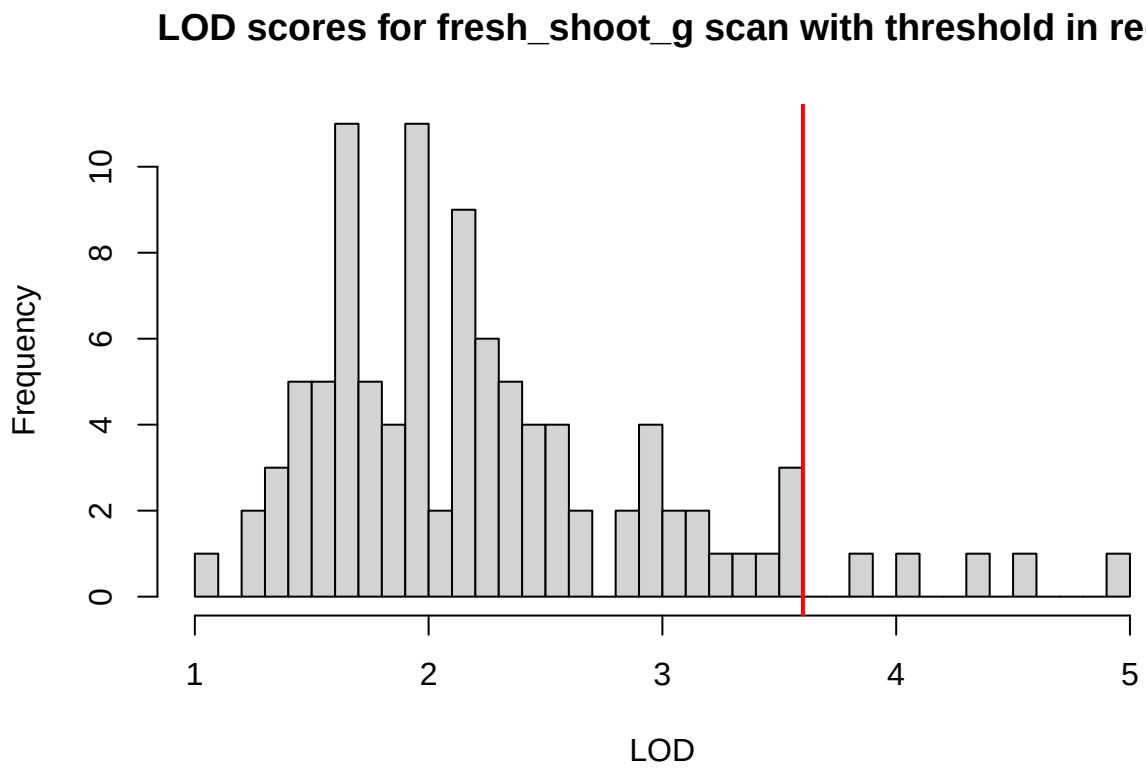


permutation test; is our LOD peak statistically significant comparing to randomly generated LOD peaks?

```
operm <- scan1perm(genoprobs = pr, pheno = test$pheno, n_perm = 100, cores=parallel::detectCores()-1)
summary(operm, alpha=c(0.2, 0.05))
```

```
## LOD thresholds (100 permutations)
##      fresh_shoot_g fresh_root_g total_fresh_biomass_g
## 0.2           2.86         2.74                  2.82
## 0.05          3.60         3.20                  3.43
```

```
hist(operm[,'fresh_shoot_g'], breaks = 50, xlab = "LOD", main = "LOD scores for fresh_shoot_g scan with
abline(v = summary(operm)[,'fresh_shoot_g'], col = 'red', lwd = 2)
```



Find peaks that has higher LOD threshold that is determined from permutation test

```
thr <- summary(operm)
find_peaks(scan1_output = out, map = map, threshold = thr, prob = 0.95, expand2markers = T)
```

```
##   lodindex             lodcolumn  chr      pos      lod    ci_lo     ci_hi
## 1        1         fresh_shoot_g ch05  44.1000 7.629985 43.05117  46.12834
## 2        2          fresh_root_g ch04 101.3153 3.974327 76.74435 105.39196
## 3        3 total_fresh_biomass_g ch05  44.0000 7.088931 42.54355  46.12834
```

Look at the markers in the range of interval (between ci_lo and ci_hi) where it showed significant LOD score for the phenotype 'fresh_shoot_g' in chromosome 5.

```
map$ch05 %>%
  as.data.frame()%>%
  filter(between(., 43.05117, 46.12834)) # Questionhere: how to get physical position of them?
```

```
##                      .
## snp1912       43.05117
## snp1913       43.05117
## cch05.loc43.1 43.10000
## cch05.loc43.2 43.20000
## cch05.loc43.3 43.30000
## cch05.loc43.4 43.40000
## cch05.loc43.5 43.50000
```

```
## snp1914       43.55879
## snp1915       43.55879
## cch05.loc43.6 43.60000
## cch05.loc43.7 43.70000
## cch05.loc43.8 43.80000
## cch05.loc43.9 43.90000
## cch05.loc44   44.00000
## cch05.loc44.1 44.10000
## cch05.loc44.2 44.20000
## cch05.loc44.3 44.30000
## cch05.loc44.4 44.40000
## cch05.loc44.5 44.50000
## snp1916       44.58975
## snp1917       44.58975
## snp1918       44.58975
## snp1919       44.58975
## snp1920       44.58975
## snp1921       44.58975
## snp1922       44.58976
## snp1923       44.58976
## snp1924       44.58976
## snp1925       44.58976
## cch05.loc44.6 44.60000
## cch05.loc44.7 44.70000
## cch05.loc44.8 44.80000
## cch05.loc44.9 44.90000
## cch05.loc45   45.00000
## snp1926       45.09737
## snp1927       45.09737
## snp1928       45.09737
## snp1929       45.09737
## cch05.loc45.2 45.20000
## cch05.loc45.3 45.30000
## cch05.loc45.4 45.40000
## cch05.loc45.5 45.50000
## cch05.loc45.6 45.60000
## cch05.loc45.7 45.70000
## cch05.loc45.8 45.80000
## cch05.loc45.9 45.90000
## cch05.loc46   46.00000
## cch05.loc46.1 46.10000
## snp1930       46.12834
## snp1931       46.12834
```

Estimate QTL effects on chromosome 5 for the fresh_shoot_g

```
c2eff <- scan1coef(pr[,"ch05"], test$pheno[,"fresh_shoot_g"]) #getting coefficient
dim(c2eff)
```

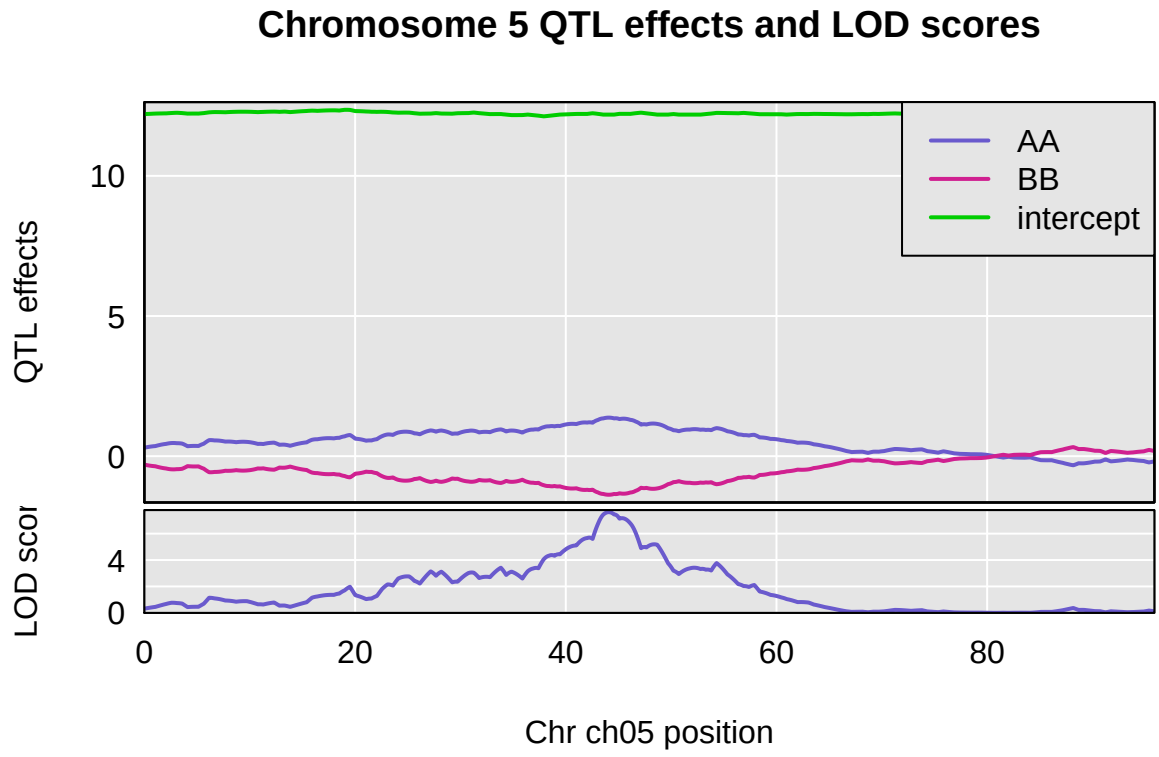
```
## [1] 1278    3
```

```
head(c2eff)
```

```
##                     AA         BB intercept
## snp1757      0.3027348 -0.3027348  12.20301
## snp1758      0.3027348 -0.3027348  12.20301
```

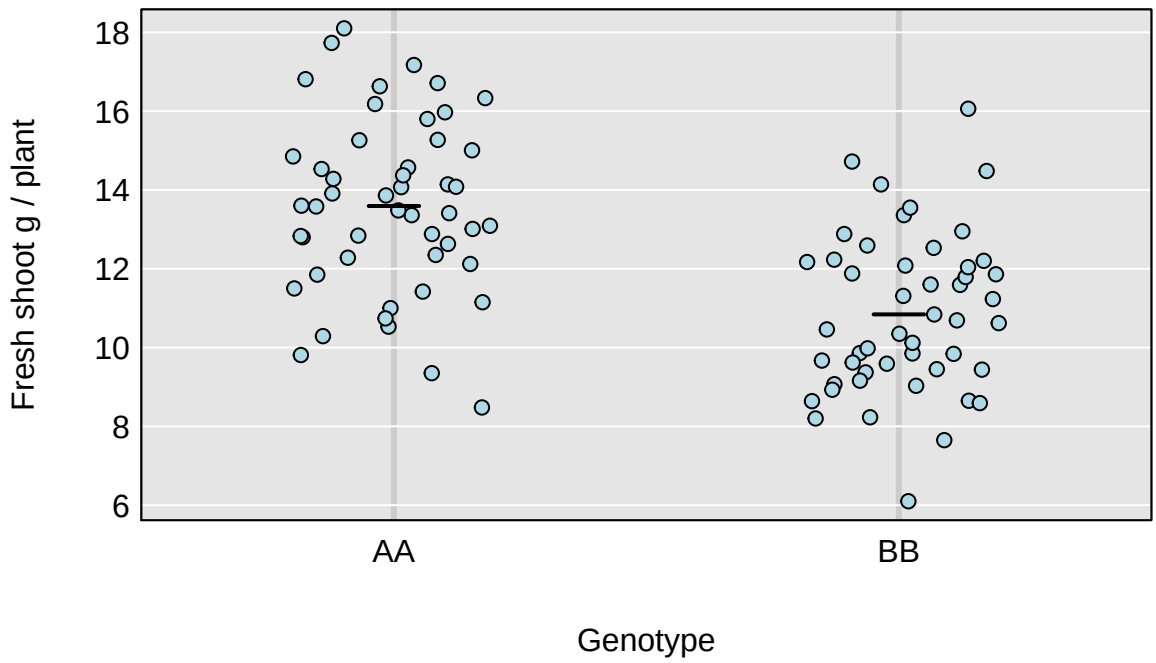
```
## cch05.loc0.1 0.3104026 -0.3104026  12.20468
## cch05.loc0.2 0.3177067 -0.3177067  12.20637
## cch05.loc0.3 0.3246068 -0.3246068  12.20810
## cch05.loc0.4 0.3310653 -0.3310653  12.20985
```

```
#plot_coef(c2eff, map, legend = "topright")
plot_coef(c2eff, map, scan1_output = out,
          main = "Chromosome 5 QTL effects and LOD scores",
          legend = "topright")
```



Plot the raw phenotypes against the genotypes at a single putative QTL position

```
g <- as.factor(maxmarg(pr, map, chr="ch05", pos=44.1000, return_char=TRUE))
plot_pxg(g, test$pheno[,"fresh_shoot_g"], ylab="Fresh shoot g / plant")
```



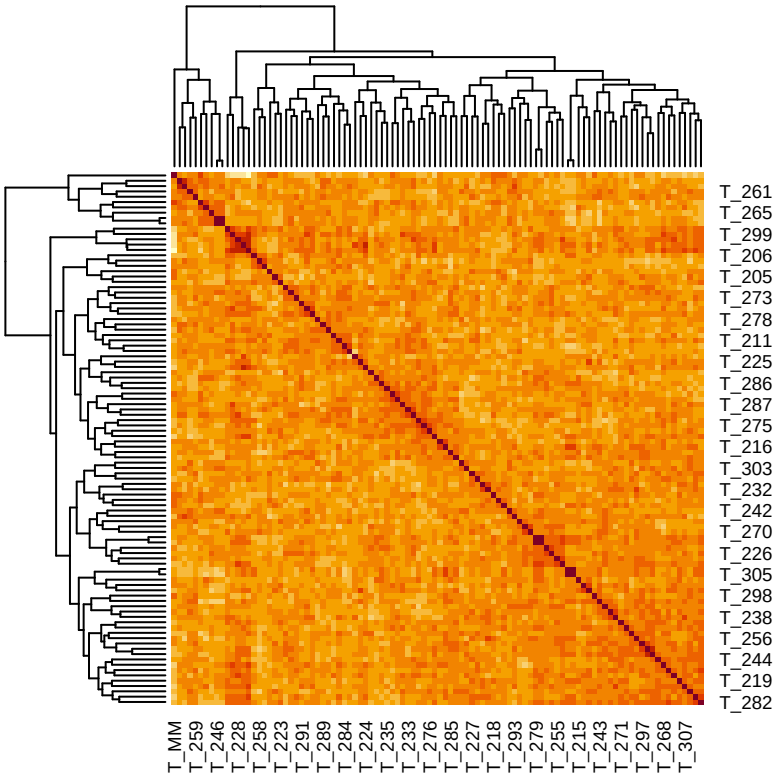
There are other methods that incorporate kinship and find LOD peak by either linear mixed model or leave one choromosome out (LOCO) method.

Calculate Kinship matrix and perform a genome scan using a linear mixed model

```
kinship <- calc_kinship(probs = pr)
kinship[1:5, 1:5]
```

```
##               T_MM        T_251     T_205     T_206     T_207
## T_MM  9.999268e-01 7.317771e-05 0.6111815 0.3825838 0.3486678
## T_251 7.317771e-05 9.999268e-01 0.3888185 0.6174162 0.6513322
## T_205 6.111815e-01 3.888185e-01 0.9964588 0.5299421 0.5616939
## T_206 3.825838e-01 6.174162e-01 0.5299421 0.9963969 0.6047783
## T_207 3.486678e-01 6.513322e-01 0.5616939 0.6047783 0.9970126
```

```
heatmap(kinship, symm = TRUE)
```



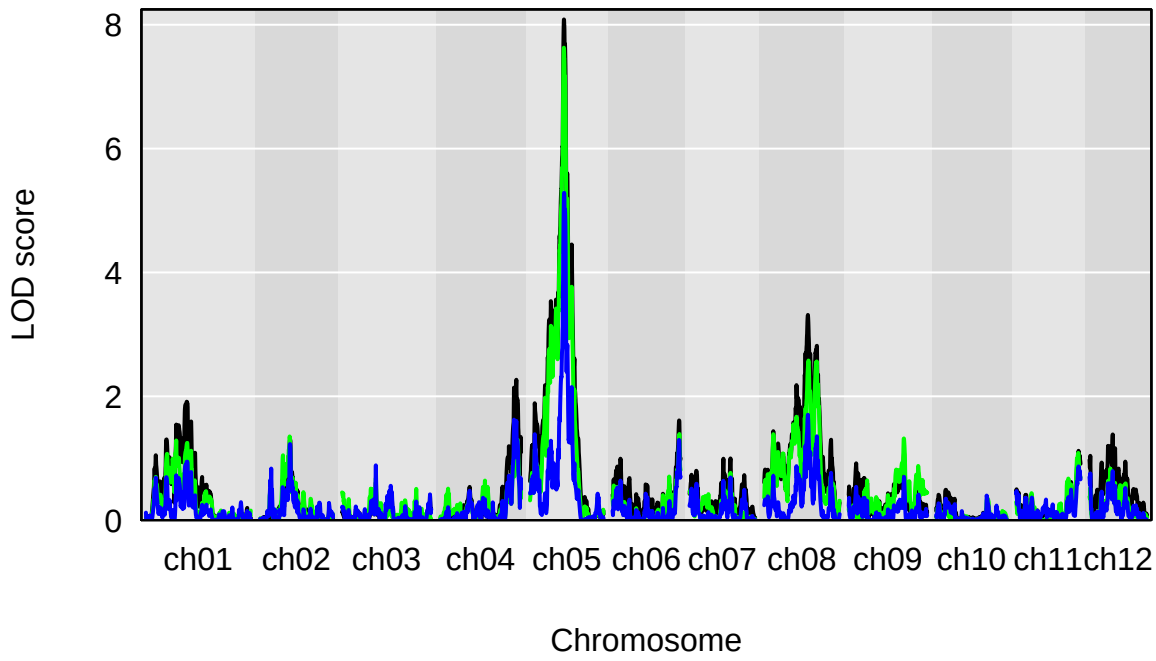

```
out_pg <- scan1(pr, test$pheno, kinship=kinship, cores=parallel::detectCores()-1)
```

Use the "leave one chromosome out" (LOCO) method

```
kinship_loco <- calc_kinship(pr, "loco")
out_pg_loco <- scan1(pr, test$pheno, kinship_loco, cores=parallel::detectCores()-1)
```

Compare LOD peaks from three different methods

```
plot_scan1(out_pg_loco, map = map, lodcolumn = "fresh_shoot_g", col = "black")
plot_scan1(out,         map = map, lodcolumn = "fresh_shoot_g", col = "green", add = TRUE)
plot_scan1(out_pg,      map = map, lodcolumn = "fresh_shoot_g", col = "blue",  add = TRUE)
```


LOD score
8
6
4
2
0
ch01
ch02
ch03
ch04
ch05
ch06
ch07
ch08
ch09
ch10
ch11
ch12
Chromosome