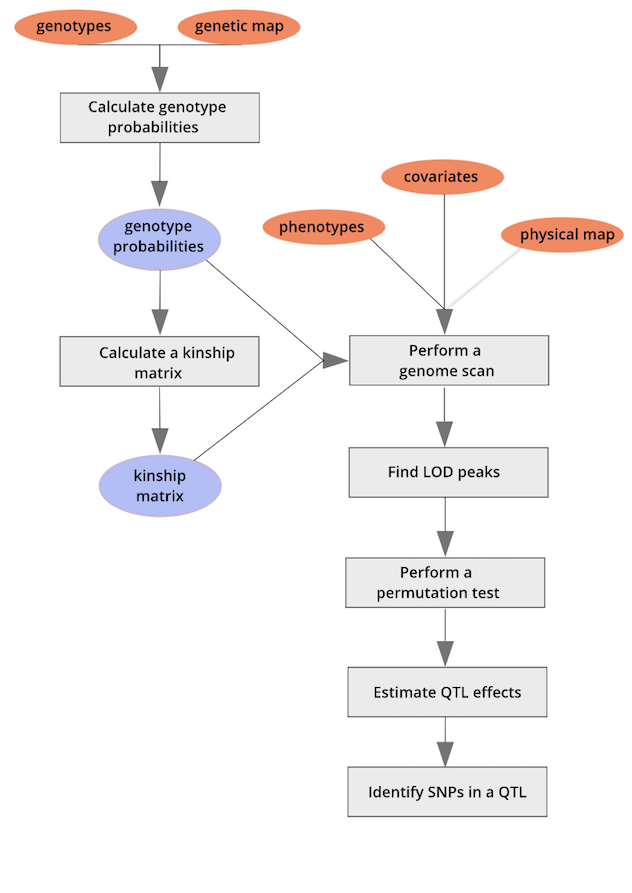
QTL analysis of behavioral traits in Diversity Outbred Mice

This tutorial will take you through the process of mapping a QTL and searching for candidate genes.

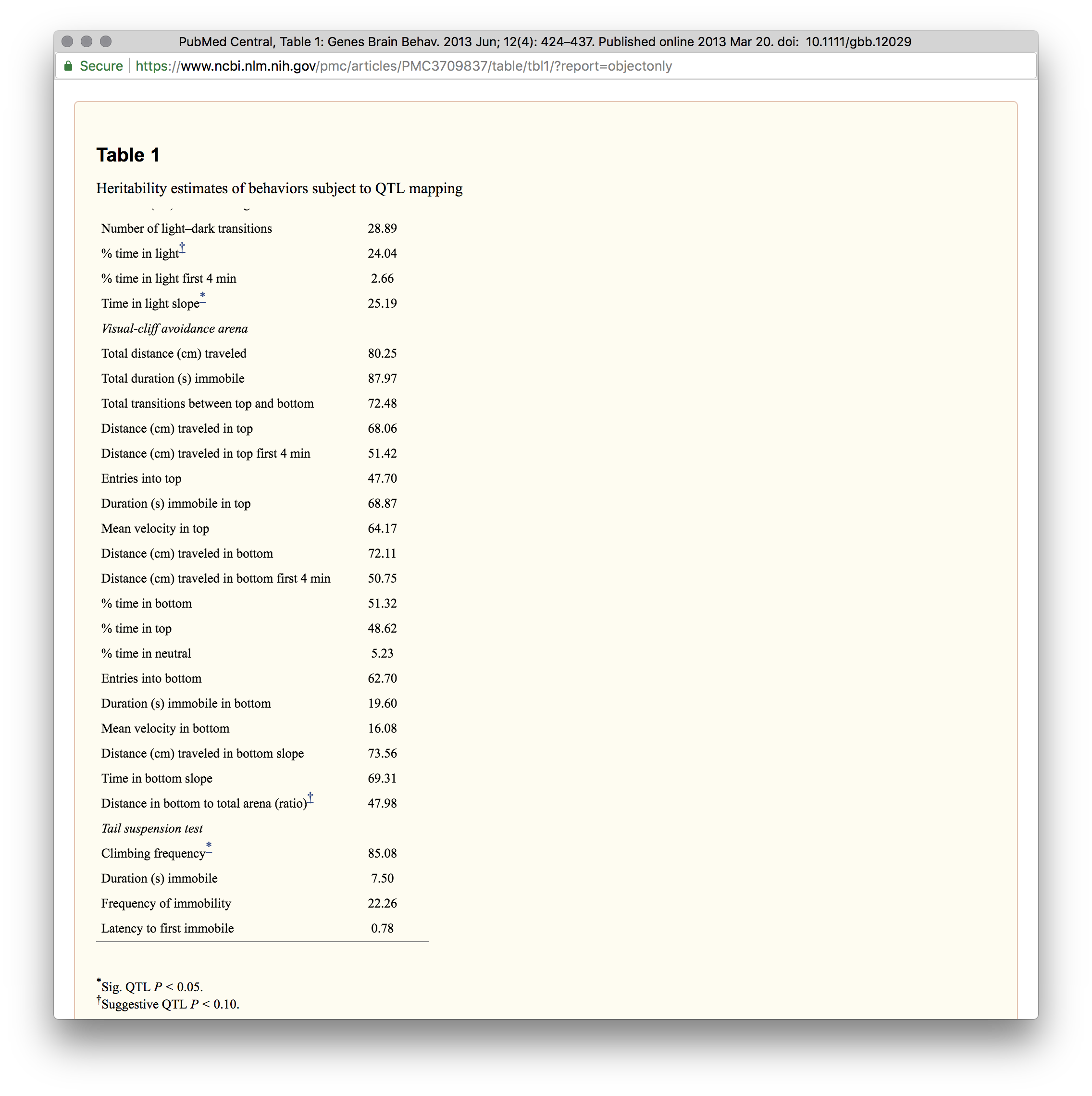
The data comes from a behavioral study in which Diversity Outbred (DO) mice were we characterized *283* male and female DO mice using open-field, light–dark box, tail-suspension and visual-cliff avoidance tests to generate 38 behavioral measures. We identified several quantitative trait loci (QTL) for these traits with support intervals ranging from 1 to 3 Mb in size. These intervals contain relatively few genes (ranging from 5 to 96). For a majority of QTL, using the founder allelic effects together with whole genome sequence data, we could further narrow the positional candidates. Several QTL replicate previously published loci. Novel loci were also identified for anxiety- and activity-related traits. Half of the QTLs are associated with wild-derived alleles, confirming the value to behavioral genetics of added genetic diversity in the DO. In the presence of wild-alleles we sometimes observe behaviors that are qualitatively different from the expected response. Our results demonstrate that high-precision mapping of behavioral traits can be achieved with moderate numbers of DO animals, representing a significant advance in our ability to leverage the mouse as a tool for behavioral genetics

There are two publications associated with these data. These are [Recla JM, et al. (2014) (Mamm Genome 25:211-222)](https://www.ncbi.nlm.nih.gov/pubmed/24700285) and [Logan RW, et al. (2014) (Genes Brain Behav 12:424-437)](https://www.ncbi.nlm.nih.gov/pubmed/23433259)

## QTL mapping workflow



### Trait heritability

Heritability estimates for the 38 behavioral measurements were calculated from the progenitor strain data using variance components from a mixed model with strain as a random effect.  Heritabilities can also be computed on measurements collected using the DO by leveraging the information containted in the kinship matrix.

### Load and explore the data

do.cross <- read\_cross2("../data/DO\_2014/DO2014.json")  
summary(do.cross)

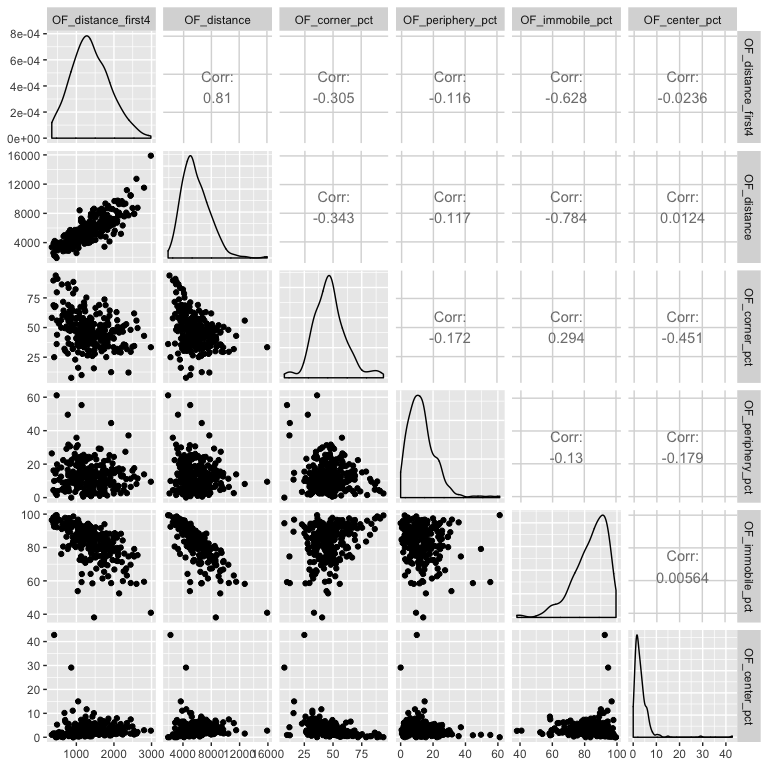
## Object of class cross2 (crosstype "do")  
##   
## Total individuals 261  
## No. genotyped individuals 261  
## No. phenotyped individuals 261  
## No. with both geno & pheno 261  
##   
## No. phenotypes 26  
## No. covariates 6  
## No. phenotype covariates 0  
##   
## No. chromosomes 20  
## Total markers 6394  
##   
## No. markers by chr:  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18   
## 448 447 396 376 359 378 348 305 321 358 314 300 306 306 248 246 216 207   
## 19 X   
## 146 369

#### The Marker Map

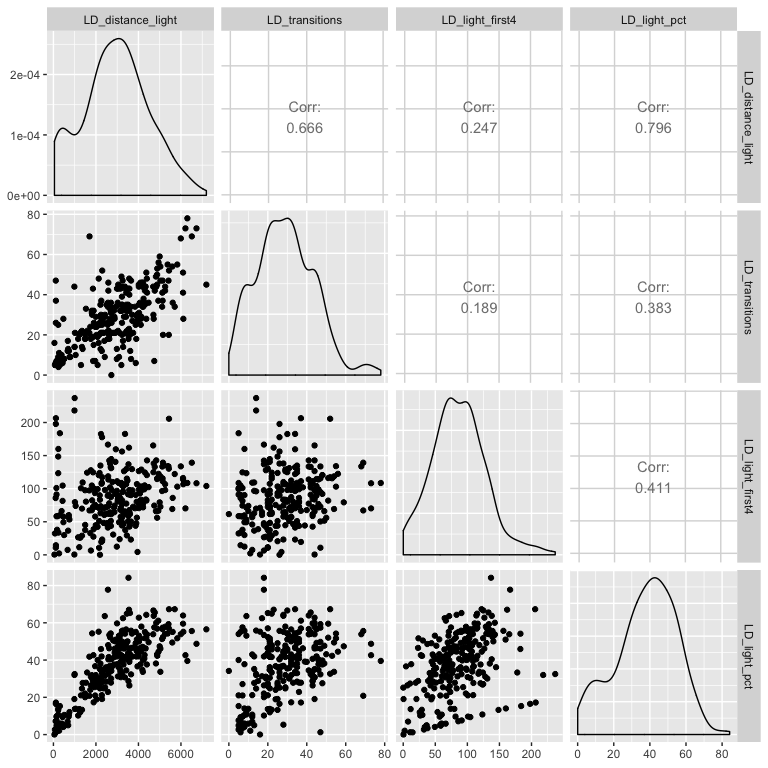
The markers are from a genotyping array called the Mouse Universal Genotyping Array (MUGA) and contains 7,856 SNP markers. Marker locations for the MUGA and other mouse arrays are available from [The Jackson Laboratory’s FTP site](ftp://ftp.jax.org/MUGA).

#### Phenotype distributions

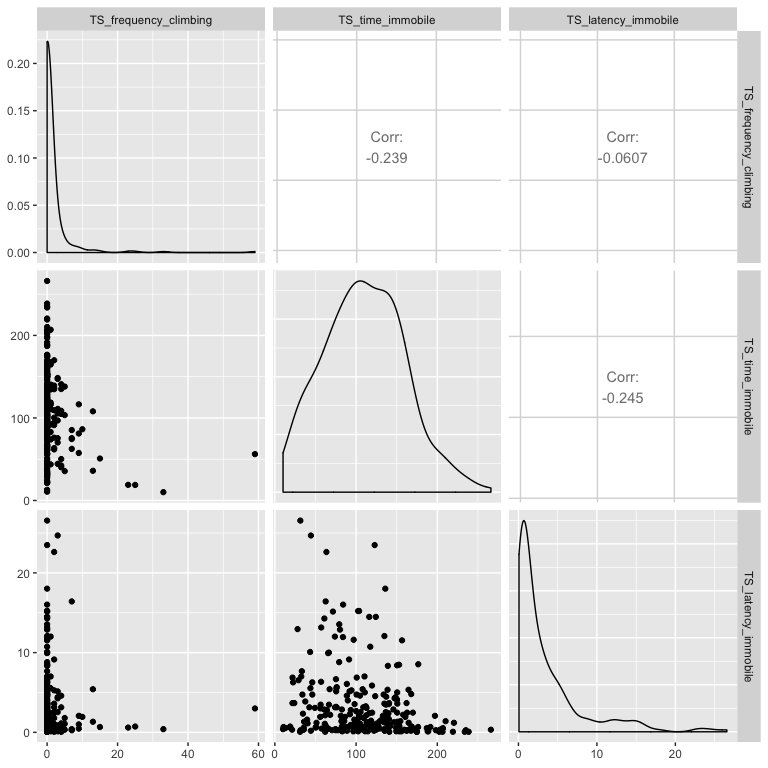
**Open field (OF)**



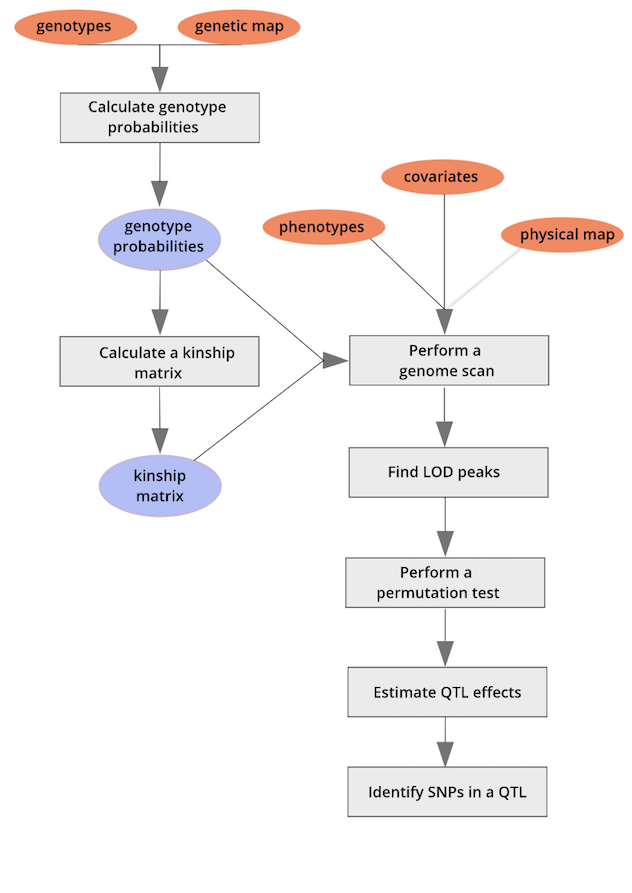
**Light Dark (LD)**



**Tail Suspension Test (TST)**



## QTL mapping of climbing frequency in the tail suspension test

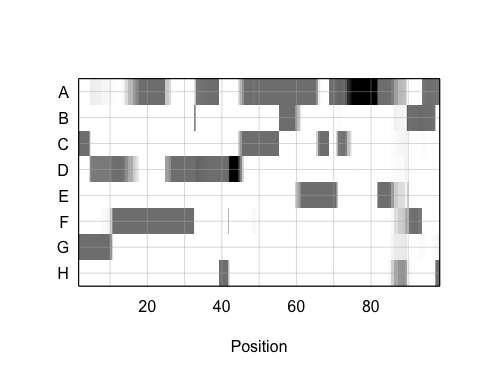


#### **Genotype (allele) probabilities**

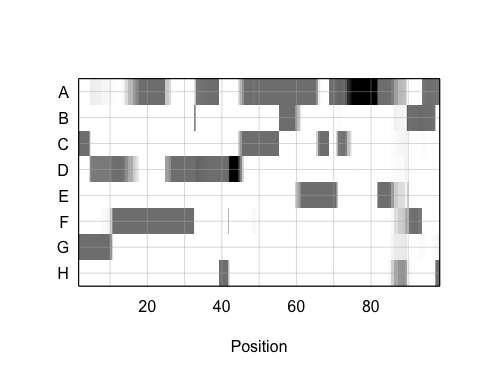
n\_cores <- 8  
probs <- calc\_genoprob(do.cross, error\_prob=0.002, map\_function="c-f", cores=n\_cores)

We now convert the genotype probabilities to haplotype dosages.

aprobs <- genoprob\_to\_alleleprob(probs, cores=n\_cores)

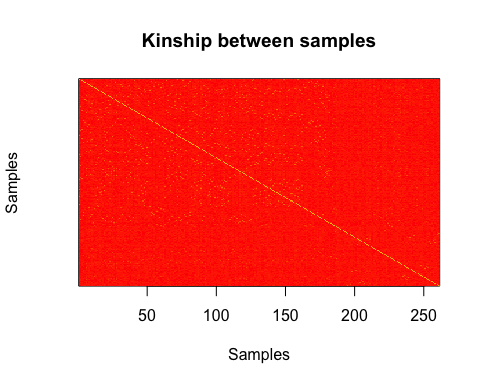
A closer look at the **eight** haplotypes for just the Chr 1 markers and couple of the DO mice. #### Sample 1 - Chr 1 

#### Sample 10 - Chr 1



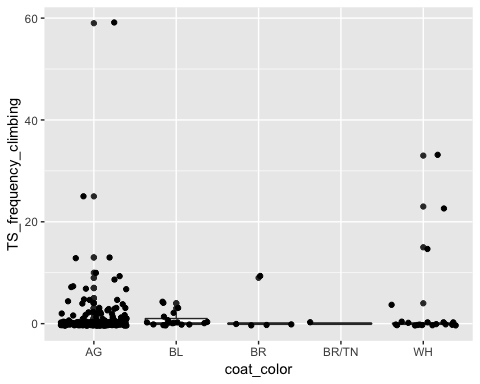
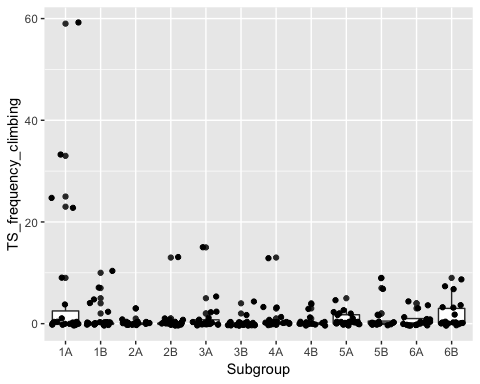
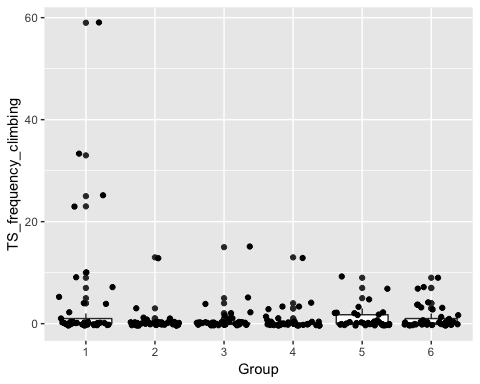
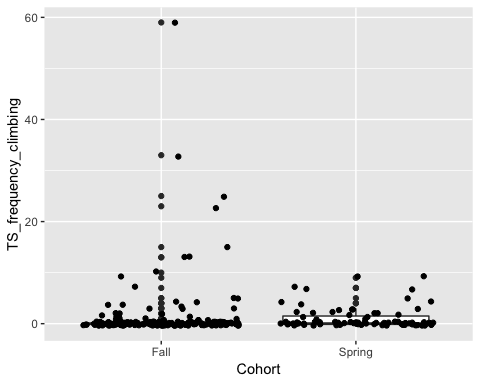
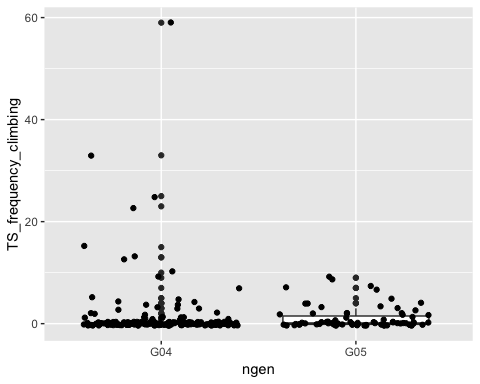
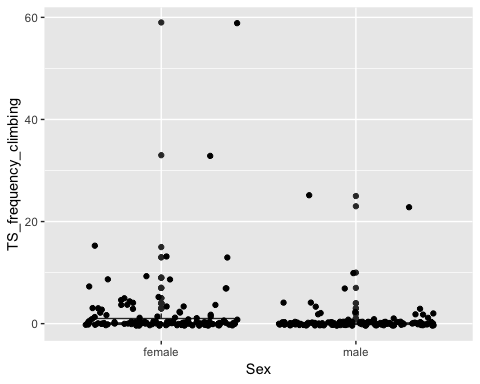
#### **Calculate a kinship matrix**

kinship <- calc\_kinship(aprobs, "loco", cores=n\_cores)  
  
image(1:nrow(kinship[[1]]), 1:ncol(kinship[[1]]), kinship[[1]][,ncol(kinship[[1]]):1], xlab = "Samples",   
 ylab = "Samples", yaxt = "n", main = "Kinship between samples",   
 breaks = 0:100/100, col = heat.colors(length(0:100) - 1))



The figure above shows kinship between all pairs of samples. White ( = 1) indicates no kinship and red ( = 0) indicates full kinship. Orange values indicate varying levels of kinship between 0 and 1. The white diagonal of the matrix indicates that each sample is identical to itself. The lighter yellow blocks off of the diagonal may indicate siblings or cousins.

#### **Covariates**



#### **Including covariates in the mapping model**

addcovar = model.matrix(~ngen+Group, data = TS.pheno)[,-1]

#### **Running a genomescan**

Before we run the mapping function, let’s look at the mapping model. At each marker on the genotyping array, we will fit a model that regresses the phenotype *LD\_light\_pct* on covariates and the founder allele probabilities.



where:

yi is the phenotype for mouse i,

βs is the effect of study cohort,

si is the study cohort for mouse i,

βj is the effect of founder allele j,

gij is the probability that mouse i carries an allele from founder j,

λi is an adjustment for kinship-induced correlated errors for mouse i,

εi is the residual error for mouse i.

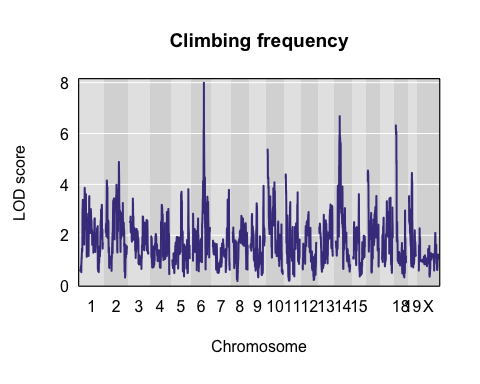
Note that this model will give us an estimate of the effect of each founder allele at each marker. There are eight founder strains that contributed to the DO, so we will get eight founder allele effects.

In order to map LD\_light\_pct, you will use the [scan1](https://github.com/rqtl/qtl2/blob/master/R/plot_scan1.R) function. To see the arguments for [scan1](https://github.com/rqtl/qtl2/blob/master/R/plot_scan1.R), you can type help(scan1). First, let’s map the *untransformed* phenotype.

index = which(colnames(pheno) == "TS\_frequency\_climbing")  
qtl.climb = scan1(genoprobs = aprobs, pheno = pheno[,index, drop = FALSE], kinship = kinship, addcovar = addcovar)

Plot of the results, all in gray except for the trait with the largest LOD score, in blue.

plot\_scan1(x = qtl.climb, map = do.cross$pmap, main = "Climbing frequency")



#### Performing a permutation test

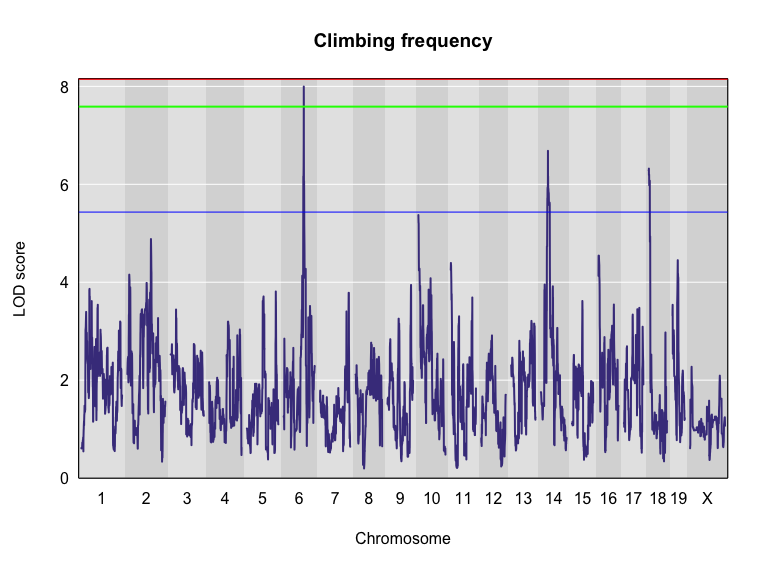
There is resonably large peak on Chr 6, with two additional peaks on Chr 14 and chr 16. Next, we must assess its statistical significance. This is most commonly done via [permutation](http://www.genetics.org/content/178/1/609.long). We advise running at least 1,000 permutations to obtain significance thresholds. In the interest of time, we perform 500 permutations here.

perms = scan1perm(genoprobs = aprobs, pheno = pheno[,index, drop = FALSE], addcovar = addcovar, n\_perm = 100)

The perms object contains the maximum LOD score from each genome scan of permuted data.

We can now add thresholds to the previous QTL plot. We use a significance threshold of p < 0.05. To do this, we select the 95th percentile of the permutation LOD distribution.

plot(x = qtl.climb, map = do.cross$pmap, main = "Climbing frequency")  
thr95 = summary(perms, alpha = 0.05)  
thr90 = summary(perms, alpha = 0.10)  
thr63 = summary(perms, alpha = 0.63)  
abline(h = c(thr95), col = "red", lwd = 3)  
abline(h = c(thr90), col = "green", lwd = 2)  
abline(h = c(thr63), col = "blue", lwd = 1)



The peak on Chr 6 is above the red significance line. #### Finding LOD peaks We can then plot the QTL scan. Note that you must provide the marker map, which we loaded earlier in the MUGA SNP data.

We can find all of the peaks above the significance threshold using the [find\_peaks](https://github.com/rqtl/qtl2/blob/master/R/find_peaks.R) function.

find\_peaks(scan1\_output = qtl.climb, map = do.cross$pmap, threshold = thr63)

## lodindex lodcolumn chr pos lod  
## 1 1 TS\_frequency\_climbing 6 97.82398 7.996058  
## 2 1 TS\_frequency\_climbing 14 36.65222 6.682980  
## 3 1 TS\_frequency\_climbing 18 4.33063 6.320311

The support interval is determined using the [Bayesian Credible Interval](http://www.ncbi.nlm.nih.gov/pubmed/11560912) and represents the region most likely to contain the causative polymorphism(s). We can obtain this interval by adding a prob argument to [find\_peaks](https://github.com/rqtl/qtl2/blob/master/R/find_peaks.R). We pass in a value of 0.95 to request a support interval that contains the causal SNP 95% of the time.

find\_peaks(scan1\_output = qtl.climb, map = do.cross$pmap, threshold = thr90, prob = 0.95)

## lodindex lodcolumn chr pos lod ci\_lo ci\_hi  
## 1 1 TS\_frequency\_climbing 6 97.82398 7.996058 95.67409 98.15002

From the output above, you can see that the support interval is 2.48 Mb wide (95.67 to 98.15 Mb). The location of the maximum LOD score is at 97.82 Mb.

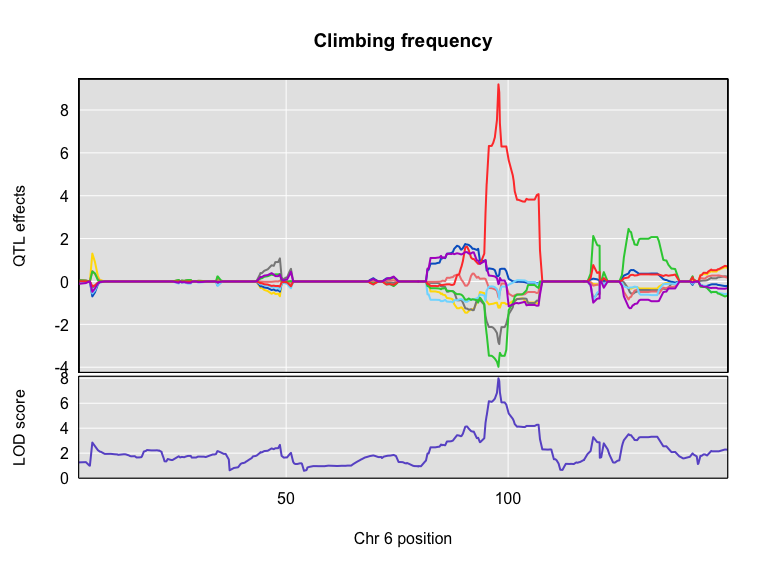
#### Estimated QTL effects

We will now zoom in on Chr 6 and look at the contribution of each of the eight founder alleles to the proportion of bone marrow reticulocytes that were micro-nucleated. Remember, the mapping model above estimates the effect of each of the eight DO founders. We can plot these effects (also called ‘coefficients’) across Chr 6 using [scan1coef](https://github.com/rqtl/qtl2/blob/master/R/scan1coef.R).

chr = 6  
coef06 = scan1blup(genoprobs = aprobs[,chr], pheno = pheno[,index, drop = FALSE], kinship = kinship[[chr]], addcovar = addcovar)

This produces an object containing estimates of each of the eight DO founder allele effect. These are the βj values in the mapping equation above.

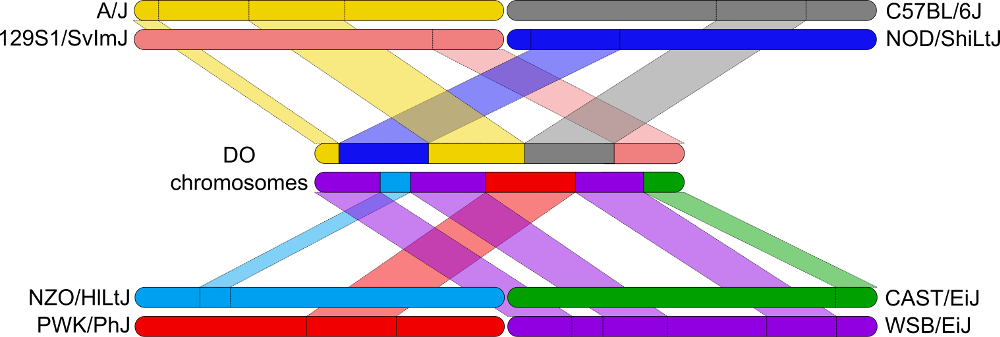
plot\_coefCC(x = coef06, map = do.cross$pmap, scan1\_output = qtl.climb, main = "Climbing frequency")



The top panel shows the eight founder allele effects (or model coefficients) along Chr 6. The founder allele effects are centerd at zero and the units are the same as the phenotype. You can see that DO mice containing the PWK/PhJ allele (red) have higher climbing frequencies relative to the other seven founders. Additional DO mice with C57BL/6J (grey) and PWK/PhJ (green) alleles have lower climbing frequncies. This means that the PWK/PhJ allele is associated with increased climbing and is potentially indicative of lower anxeity and higher exploration. The bottom panel shows the LOD score, with the support interval for the peak shaded blue.

#### SNP Association Mapping

At this point, we have a 2.50 Mb wide support interval that contains a polymorphism(s) that influences climbing behavior. Next, we will impute the DO founder sequences onto the DO genomes. The [Sanger Mouse Genomes Project](http://www.sanger.ac.uk/resources/mouse/genomes/) has sequenced the eight DO founders and provides SNP, insertion-deletion (indel), and structural variant files for the strains (see [Baud et.al., Nat. Gen., 2013](http://www.nature.com/ng/journal/v45/n7/full/ng.2644.html)). We can impute these SNPs onto the DO genomes and then perform association mapping. The process involves several steps and I have provided a function to encapsulate the steps. To access the Sanger SNPs, we use a SQLlite database provided by [Karl Broman](https://github.com/kbroman). You should have downloaded this during [Setup](../setup.md). It is available from the [JAX FTP site](ftp://ftp.jax.org/dgatti/CC_SNP_DB/cc_variants.sqlite), but the file is 3 GB, so it may take too long to download right now.



Association mapping involves imputing the founder SNPs onto each DO genome and fitting the mapping model at each SNP. At each marker, we fit the following model:



where:

yi is the phenotype for mouse i,

βs is the effect of study cohort,

si is the study cohort for mouse i,

βm is the effect of adding one allele at marker m,

gim is the allele call for mouse i at marker m,

λi is an adjustment for kinship-induced correlated errors for mouse i,

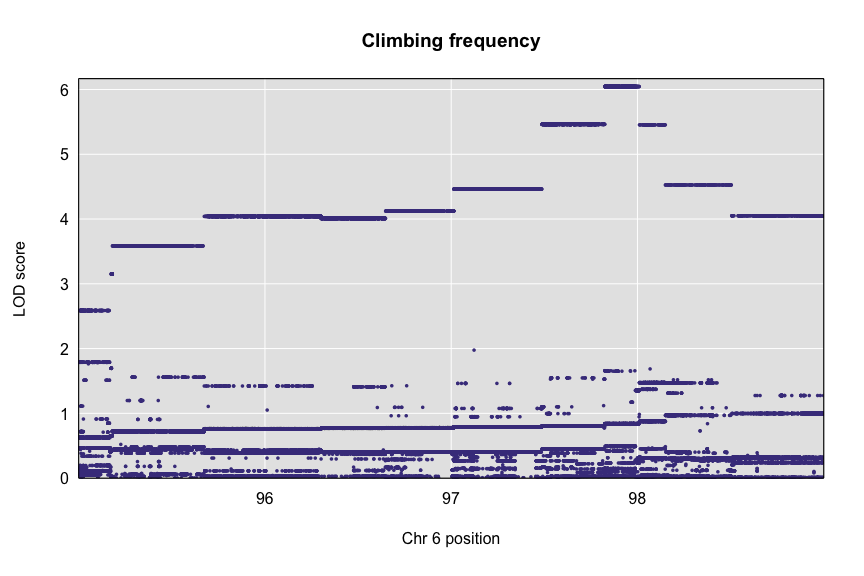
εi is the residual error for mouse i.

We can call [scan1snps](https://github.com/rqtl/qtl2/blob/master/R/scan1snps.R) to perform association mapping in the QTL interval on Chr 6. We first create variables for the chromosome and support interval where we are mapping. We then create a function to get the SNPs from the founder SNP database The path to the SNP database (snpdb\_file argument) points to the data directory on your computer. Note that it is important to use the keep\_all\_snps = TRUE in order to return all SNPs.

chr = 6  
start = 95  
end = 99  
query\_func = create\_variant\_query\_func("../data/cc\_variants.sqlite")  
assoc = scan1snps(genoprobs = aprobs[,chr], map = do.cross$pmap, pheno = pheno[,index,drop = FALSE], kinship = kinship, addcovar = addcovar, query\_func = query\_func, chr = chr, start = start, end = end, keep\_all\_snps = TRUE)

The assoc object is a list containing two objects: the LOD scores for each unique SNP and a snpinfo object that maps the LOD scores to each SNP. To plot the association mapping, we need to provide both objects to the [plot\_snpasso](https://github.com/rqtl/qtl2/blob/master/R/plot_snpasso.R) function.

plot\_snpasso(scan1output = assoc$lod, snpinfo = assoc$snpinfo, main = "Climbing frequency")



This plot shows the LOD score for each SNP in the QTL interval. The SNPs occur in “shelves” because all of the SNPs in a haplotype block have the same founder strain pattern. The SNPs with the highest LOD scores are the ones for which CAST/EiJ contributes the alternate allele.

We can add a plot containing the genes in the QTL interval using the plot\_genes function. We get the genes from another SQLlite database created by [Karl Broman](https://github.com/kbroman) called mouse\_genes.sqlite. You should have downloaded this from the [JAX FTP Site](ftp://ftp.jax.org/dgatti/CC_SNP_DB/mouse_genes.sqlite) during [Setup](../setup.md).

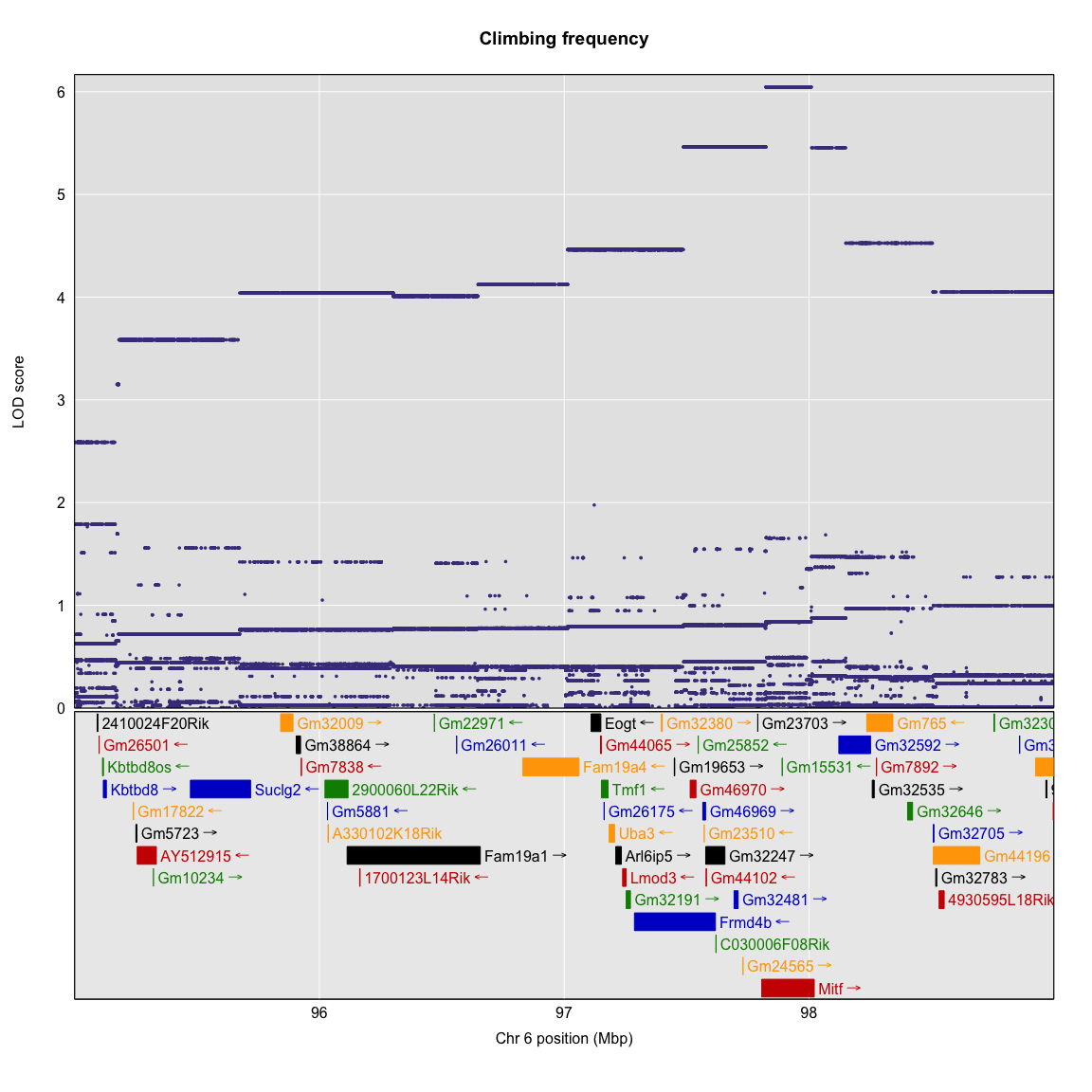
First, we must query the database for the genes in the interval. The path of the first argument points to the data directory on your computer.

query\_genes = create\_gene\_query\_func(dbfile = "../data/mouse\_genes.sqlite", filter = "source='MGI'")  
genes = query\_genes(chr, start, end)  
head(genes)

## chr source type start stop score strand phase  
## 1 6 MGI gene 95.09201 95.09508 NA <NA> NA  
## 2 6 MGI gene 95.10020 95.10031 NA - NA  
## 3 6 MGI gene 95.11401 95.11795 NA - NA  
## 4 6 MGI gene 95.11724 95.12979 NA + NA  
## 5 6 MGI pseudogene 95.23992 95.24061 NA - NA  
## 6 6 MGI pseudogene 95.25030 95.25451 NA + NA  
## ID Name Parent  
## 1 MGI:MGI:1920900 2410024F20Rik <NA>  
## 2 MGI:MGI:5456278 Gm26501 <NA>  
## 3 MGI:MGI:1919467 Kbtbd8os <NA>  
## 4 MGI:MGI:2661430 Kbtbd8 <NA>  
## 5 MGI:MGI:5010007 Gm17822 <NA>  
## 6 MGI:MGI:3647796 Gm5723 <NA>  
## Dbxref  
## 1 GenBank:AK010580  
## 2 ENSEMBL:ENSMUSG00000093999  
## 3 VEGA:OTTMUSG00000057788,NCBI\_Gene:102634268,ENSEMBL:ENSMUSG00000085413  
## 4 VEGA:OTTMUSG00000029697,NCBI\_Gene:243574,ENSEMBL:ENSMUSG00000030031  
## 5 VEGA:OTTMUSG00000057789,NCBI\_Gene:100328567,ENSEMBL:ENSMUSG00000107934  
## 6 VEGA:OTTMUSG00000057790,NCBI\_Gene:435912,ENSEMBL:ENSMUSG00000107900  
## mgiName  
## 1 RIKEN cDNA 2410024F20 gene  
## 2 predicted gene%2c 26501  
## 3 kelch repeat and BTB (POZ) domain containing 8%2c opposite strand  
## 4 kelch repeat and BTB (POZ) domain containing 8  
## 5 predicted gene%2c 17822  
## 6 predicted gene 5723  
## bioType Alias  
## 1 unclassified gene <NA>  
## 2 snRNA gene <NA>  
## 3 antisense lncRNA gene <NA>  
## 4 protein coding gene <NA>  
## 5 pseudogene <NA>  
## 6 pseudogene <NA>

The genes object contains annotation information for each gene in the interval.

Next, we will create a plot with two panels: one containing the association mapping LOD scores and one containing the genes in the QTL interval. We do this by passing in the genes argument to [plot\_snpasso](https://github.com/rqtl/qtl2/blob/master/R/plot_snpasso.R).



#### Identification of Candidate Genes

One strategy for finding genes related to a phenotype is to search for genes with expression QTL (eQTL) in the same location. Ideally, we would have brain gene expression data in the DO mice from this experiment. Unfortunately, we did not collect this data. However, we have brain (Striatum) gene expression for a separate set of naive DO mice . We searched for genes in the QTL interval that had an eQTL in the same location. Then, we looked at the pattern of founder effects to see if PWK/PhJ stood out in addition to C57BL/6J and CAST/EiJ showing the same pattern. We found one gene that met these criteria.

