Rice Quantitative Trait Loci (QTL): An Annotated Guide

Robert Gotwals, Computational Science Educator, NCSSM

Last compiled on August 06, 2023

About this *Guide*

The purpose of this guide is to provide a line-by-line, code chunk-by-code chunk description of the R code needed to perform a quantitative trait loci (QTL) analysis of breeding data between two species of rice (*Oryza sativa*): *Curinga* x *O. rufipogon*. This *Guide* does not provide a description of QTL analyses; that is found in the curricular materials. This *Guide* will annotate the required code for conducting the analyses.

This *Guide* assumes that you have downloaded and installed the R software (https://cran.rstudio.com/ (https://cran.rstudio.com/)) and the interface to R, RStudio (https://posit.co/download/rstudio-desktop/ (https://posit.co/download/rstudio-desktop/)). It also assumes that, in R, you have installed a "package" called "qtl". To install this, run the command "install.packages("qtl") at the console command line. You only have to do tis once.

Initial setup

There are three steps here:

1. Load the "qtl" package/library. This assumes that you have already installed the package, a one-time process.
2. Remove any previous items in memory. The "Global Environment" window on the top righthand side of RStudio should be empty after running this command.
3. The QTL dataset for this analyses is large by R standards, but small by genomics standards. Regardless, you need to set the system environment with enough memory to handle the data.

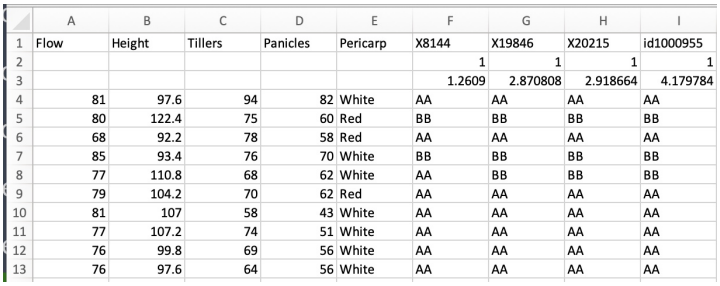
You should also add some documentation to your file by using an asterisk. Well-documented code is really important if you want to keep your job as a programmer/data scientist!

```
# Your name
# Today's date
# Rice QTL analysis
#
library(qtl)
rm(list=ls())
Sys.setenv(VROOM_CONNECTION_SIZE="500000")
#
```

Load Data

Now we are ready to load the data. The data has been upload to a server maintained by the Department of Chemistry at NCSSM, as a "comma-separated values" (csv) file. The command read.cross states that you are reading a CSV file located at http://chemistry.ncssm.edu (http://chemistry.ncssm.edu), in the data/gwas directory. The file name is CuRUFCSSL_QTL.csv. Then, the code states that there are three genotypes coded in the data: AA, H (heterozygous), and BB. Some of the genotypes are missing, and those are indicated by the

na.strings command, with a space between the quotes. Finally, the code states that there are two different alleles, A and B. It is likely that you will receive a warning, but this will not have any impact on your analyses. You will also receive a brief summary of the cross.



	A	B	C	D	E	F	G	H	I
1	Flow	Height	Tillers	Panicles	Pericarp	X8144	X19846	X20215	id1000955
2						1	1	1	1
3						1.2609	2.870808	2.918664	4.179784
4	81	97.6	94	82	White	AA	AA	AA	AA
5	80	122.4	75	60	Red	BB	BB	BB	BB
6	68	92.2	78	58	Red	AA	AA	AA	AA
7	85	93.4	76	70	White	BB	BB	BB	BB
8	77	110.8	68	62	White	AA	BB	BB	BB
9	79	104.2	70	62	Red	AA	AA	AA	AA
10	81	107	58	43	White	AA	AA	AA	AA
11	77	107.2	74	51	White	AA	AA	AA	AA
12	76	99.8	69	56	White	AA	AA	AA	AA
13	76	97.6	64	56	White	AA	AA	AA	AA

Screenshot of QTL data.

```
cross <- read.cross("csv", file="http://chemistry.ncssm.edu/data/gwas/CuRUFCSSL_QTL.csv", genotypes = c("AA","H", "BB"), na.strings=" ", alleles = c("A", "B"))
```

```
## Warning in read.cross.csv(dir, file, na.strings, genotypes, estimate.map, : The following unexpected genotype codes were treated as missing.
##     |NA|
```

```
##  --Read the following data:
##   256  individuals
##   1769  markers
##   5  phenotypes
##  --Cross type: f2
```

It is also helpful to display the names of the phenotypes (pay attention to case!). Finally, it's instructive to request a summary of the cross data, which provides a detailed analyses of the number of crosses, descriptions of the phenotypes and genotypes, etc.

```
names(cross$pheno)
```

```
## [1] "Flow"     "Height"   "Tillers"  "Panicles" "Pericarp"
```

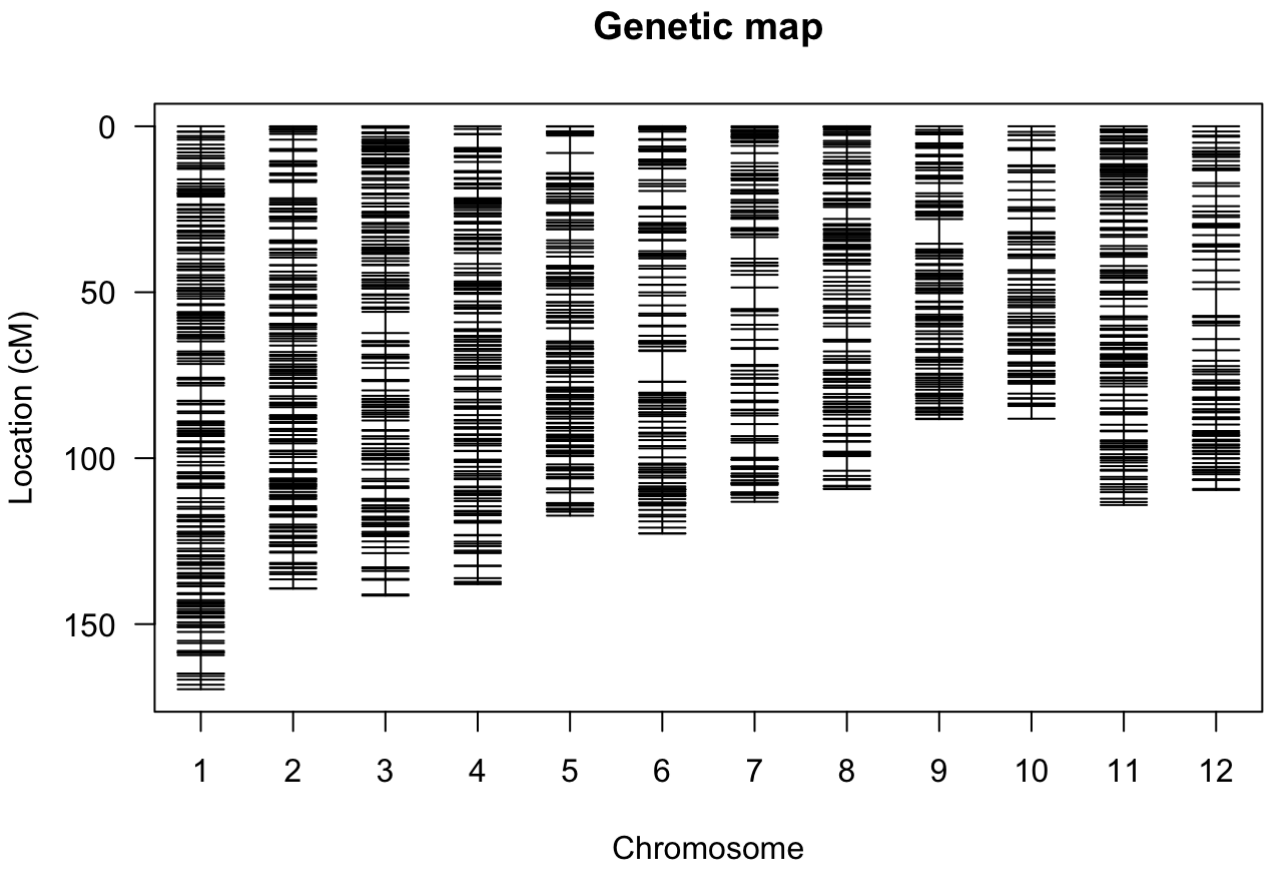
```
summary(cross)
```

```
##     F2 intercross
##
##     No. individuals:    256
##
##     No. phenotypes:     5
##     Percent phenotyped: 100 100 100 100 100
##
##     No. chromosomes:    12
##         Autosomes:      1 2 3 4 5 6 7 8 9 10 11 12
##
##     Total markers:      1769
##     No. markers:        213 183 185 174 158 133 111 141 127 89 152 103
##     Percent genotyped:  27.2
##     Genotypes (%):      AA:86.9  AB:0.7  BB:12.4  not BB:0.0  not AA:0.0
```

Genetic Markers

Next, we want to see a map of the genetic markers, and, even though this is a relatively snmall dataset, there are almost 1800 markers. The **plot.map** command shows all of the markers by chromosome. Notice that they are not evenly spaced!

```
plot.map(cross)
```



Running preliminary files

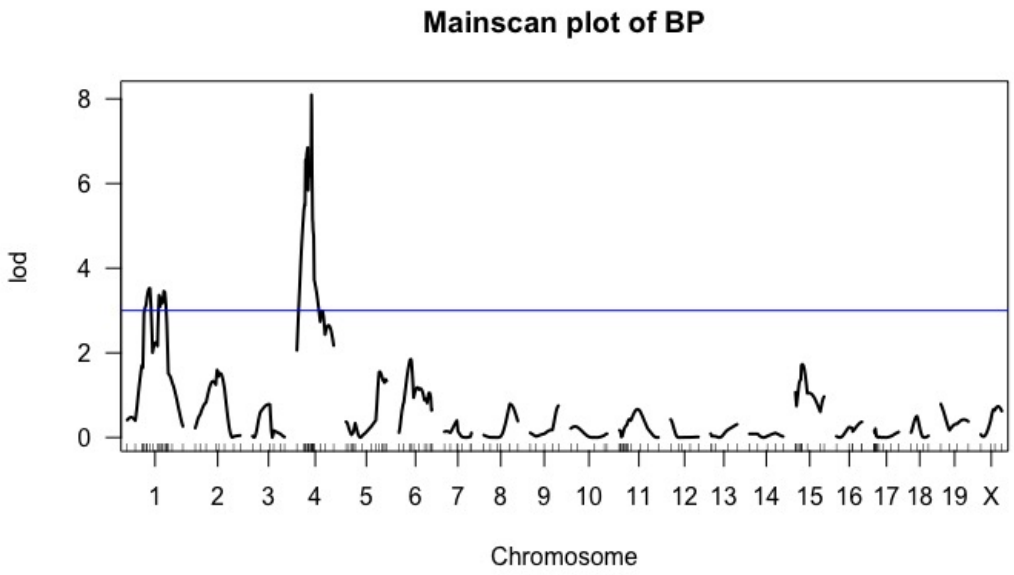
Two preliminary files need to be generated before performing the main analysis – the "mainscan" – of your data. Both of these use a machine learning method known as Hidden Markov to check the actual data against what the actual data should be, a check on genotyping errors. The options for both are the same, and are described below for the curious reader! These notes come from the QTL package documentation.

1. calc.genoprob: Uses the hidden Markov model technology to calculate the probabilities of the true underlying genotypes given the observed multipoint marker data, with possible allowance for genotyping errors.
    a. step: Maximum distance (in cM) between positions at which the simulated genotypes will be drawn, though for step=0, genotypes are drawn only at the marker locations.
    b. off.end: Distance (in cM) past the terminal markers on each chromosome to which the genotype simulations will be carried.
    c. error.prob: Assumed genotyping error rate used in the calculation of the penetrance Pr(observed genotype | true genotype).
    d. map.function: ndicates whether to use the Haldane, Kosambi, Carter-Falconer, or Morgan map function when converting genetic distances into recombination fractions.
    e. stepwidth: indicates whether the intermediate points should with fixed or variable step sizes
2. sim.genoprob: Uses the hidden Markov model technology to simulate from the joint distribution Pr(g | O) where g is the underlying genotype vector and O is the observed multipoint marker data, with possible allowance for genotyping errors. The option "n.draws" describes the number of simulated probabilities to calculate.

```
cross <- calc.genoprob(cross, step=2.0, off.end=0.0, error.prob=1.0e-4, map.function= "h
aldane", stepwidth ="fixed")
cross <- sim.geno(cross, step=2.0, off.end=0.0, error.prob=1.0e-4, map.function= "haldan
e", stepwidth ="fixed", n.draws=16)
```

Running a mainscan for plant height

Now we are ready for the main goal of the analyses: to find where on one or more chromosomes there might be genes that are responsible for a specific trait, or phenotype. The graphic below comes from mouse data, and we are looking to see where genes that control blood pressure (BP) might be located. On the x-axis, we see the 19 chromosomes of a mouse, as well as the X-chromosome. On the y-axis is a LOD score. From the National Human Genome Research Institute (https://www.genome.gov/ (https://www.genome.gov/)): "A LOD (short for "logarithm of the odds") score is a statistical estimate of the relative probability that two loci (e.g., a disease-associated gene and another sequence of interest, such as a variant or another gene) are located near each other on a chromosome and are therefore likely to be inherited together."



Screenshot of a mainscan graphic.

For a LOD score to be significant, we typically use a cutoff of 3. There is an obvious peak on Chromosome 4, so someone hunting for the BP gene would focus most of their attention there. There is also some activity on Chromosome 1, so blood pressure is a polygenetic – more than one gene – trait. Attention would also need to be paid to Chromosome 1.

For this analyses, recall that there are four numerical phenotypes, one of them being plant height. This example demonstrates that analyses. You will then have the opportunity to analyze the other three.

The command to do a mainscan is **scanone*. We are scanning the cross data, and the phenotype of interest is in Column 2 of the dataset. We are using a simple “normal” model and the “expectation-maximization” (em) method. There are, as you might suspect, different algorithms that could be applied to this analyses, but EM is the most common.

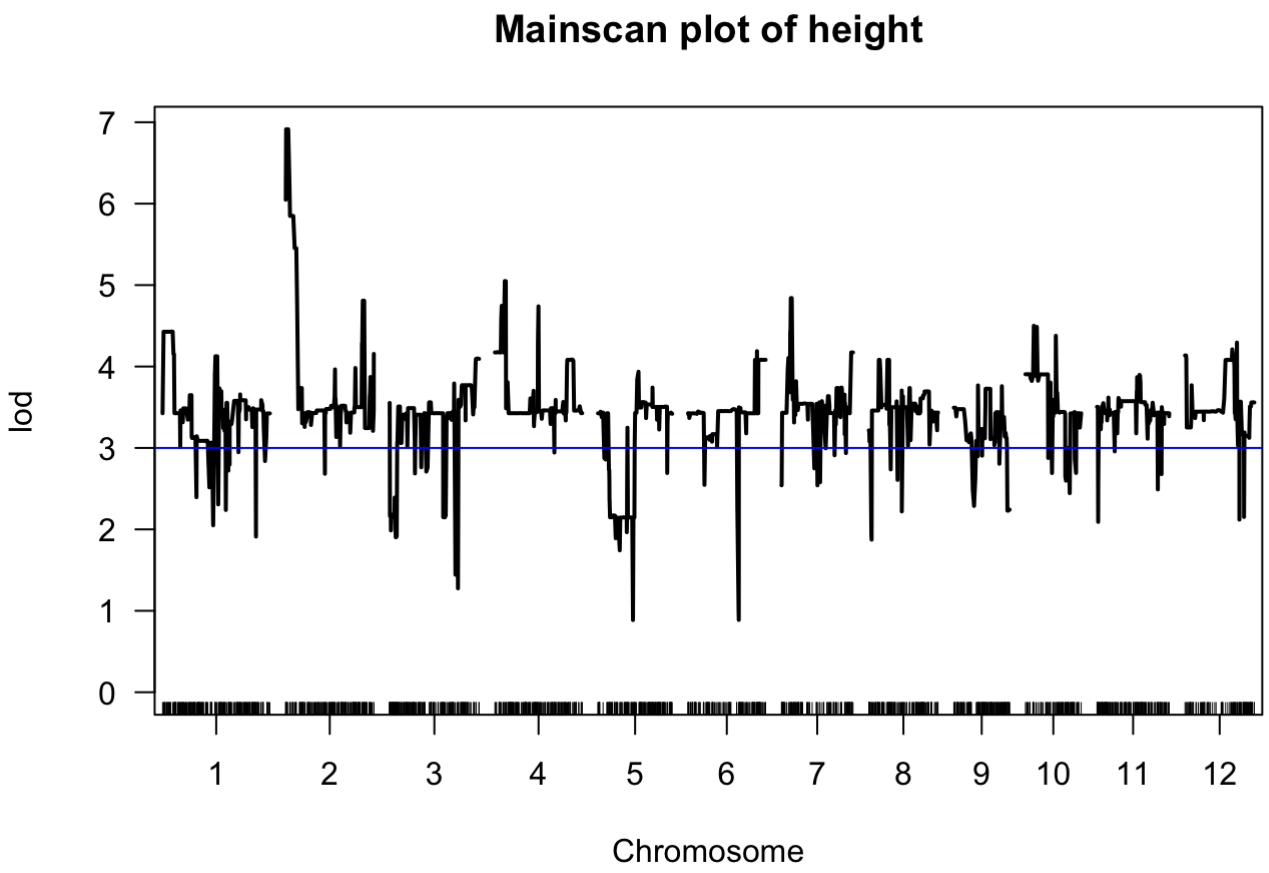
There is a second analyses we can do, called a **permutation** test. This test basically tears the data apart and puts it back together. In other words,“a permutation tests shuffles genotypes and phenotypes, essentially breaking the relationship between the two.” (https://smcclatchy.github.io/mapping/06-perform-perm-test/ (https://smcclatchy.github.io/mapping/06-perform-perm-test/)). We specify the number of permutations to run, in this case 100. For a more thorough analyses, one would run 500, 1000, or more permutations. 100 is enough for this particular dataset.

Once we run the scanones, we can plot the results. A simple **plot** command does the trick, and we can add a graph title using the **main** command. We might also want to add a threshold line at a LOD value of 3 If there are no significant QTLs, no line will be plotted as there are no peaks above that score.

```
cross.scanheight <- scanone(cross, pheno.col=2, model = "normal", method="em")
cross.scanheight.perm <- scanone(cross, pheno.col=2, model="normal", method="em", n.perm
=100)
```

```
## Permutation 5
## Permutation 10
## Permutation 15
## Permutation 20
## Permutation 25
## Permutation 30
## Permutation 35
## Permutation 40
## Permutation 45
## Permutation 50
## Permutation 55
## Permutation 60
## Permutation 65
## Permutation 70
## Permutation 75
## Permutation 80
## Permutation 85
## Permutation 90
## Permutation 95
## Permutation 100
```

```
plot(cross.scanheight, main="Mainscan plot of height")
lodline <-3
abline(h=lodline, col="blue")
```



If it is the case that you have one or more significant QTLs – those with a LOD score of 3 or greater – you might want to look at effect plots. To do that, you first need to look at a summary of your QTLs. The **summary** command will show you that. The **alpha** option says only look at the QTLs that are 95% significant. You might need to change this number to 90% (0.10) or lower.

The **summary** command will show you the ID of the closest marker, the chromosome number, the location in centiMorgans (cM), and the LOD score.You will need that information for the next step.

```
summary(cross.scanheight, perm=cross.scanheight.perm, alpha=0.05)
```

```
##           chr    pos  lod
## c1.loc14    1  15.26 4.43
## c2.loc2     2   3.12 6.91
## c3.loc138   3 139.88 4.10
## c4.loc16    4  16.24 5.05
## X6851172    6 110.02 4.19
## c7.loc16    7  18.62 4.84
## c8.loc16    8  17.47 4.08
## X10099158  10  17.25 4.50
## X12852964  12  82.58 4.29
```

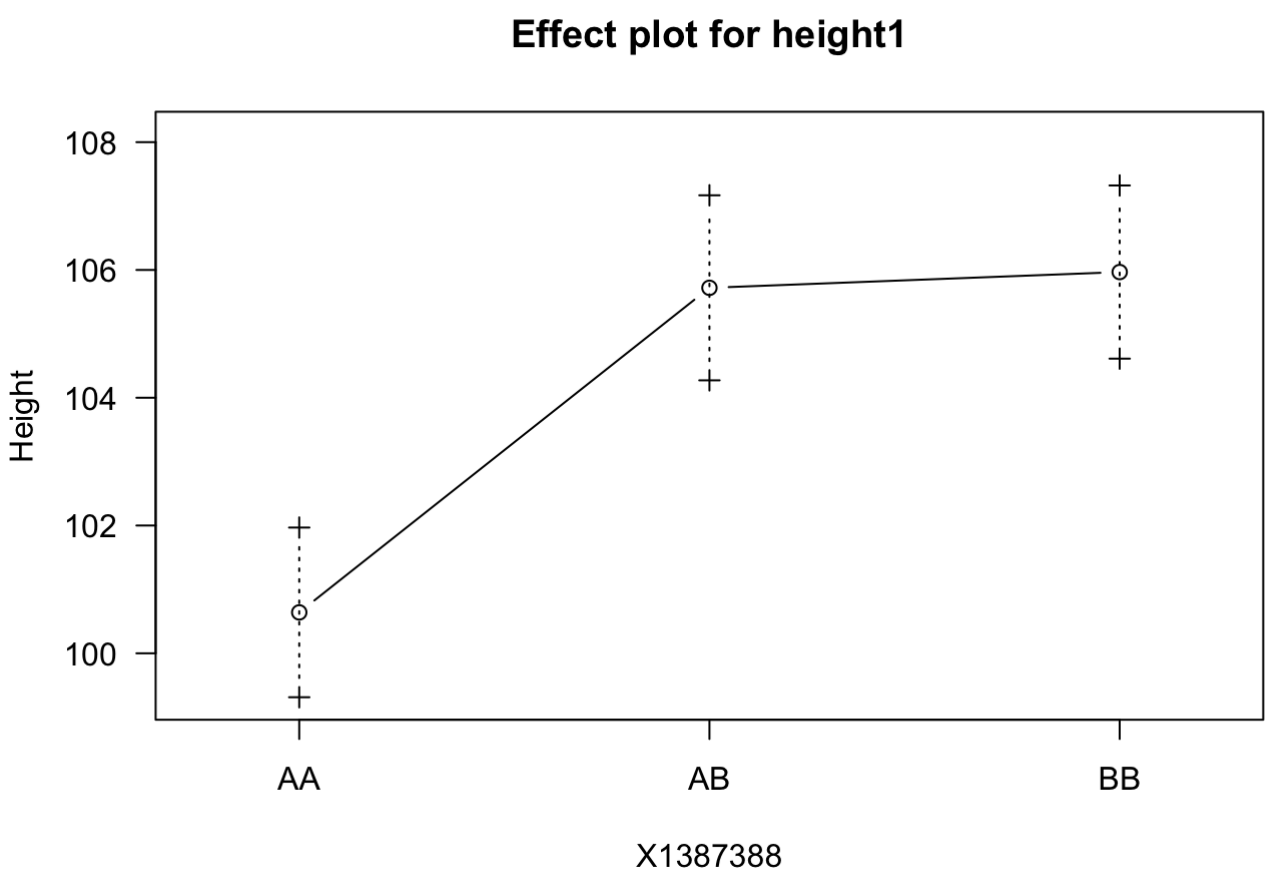
```
summary(cross.scanheight, perm=cross.scanheight.perm)
```

```
##           chr    pos  lod
## c1.loc14    1  15.26 4.43
## c2.loc2     2   3.12 6.91
## c3.loc138   3 139.88 4.10
## c4.loc16    4  16.24 5.05
## c5.loc64    5  65.02 3.94
## X6851172    6 110.02 4.19
## c7.loc16    7  18.62 4.84
## c8.loc16    8  17.47 4.08
## X9469699    9  39.65 3.77
## X10099158  10  17.25 4.50
## X11465012  11  68.91 3.90
## X12852964  12  82.58 4.29
```

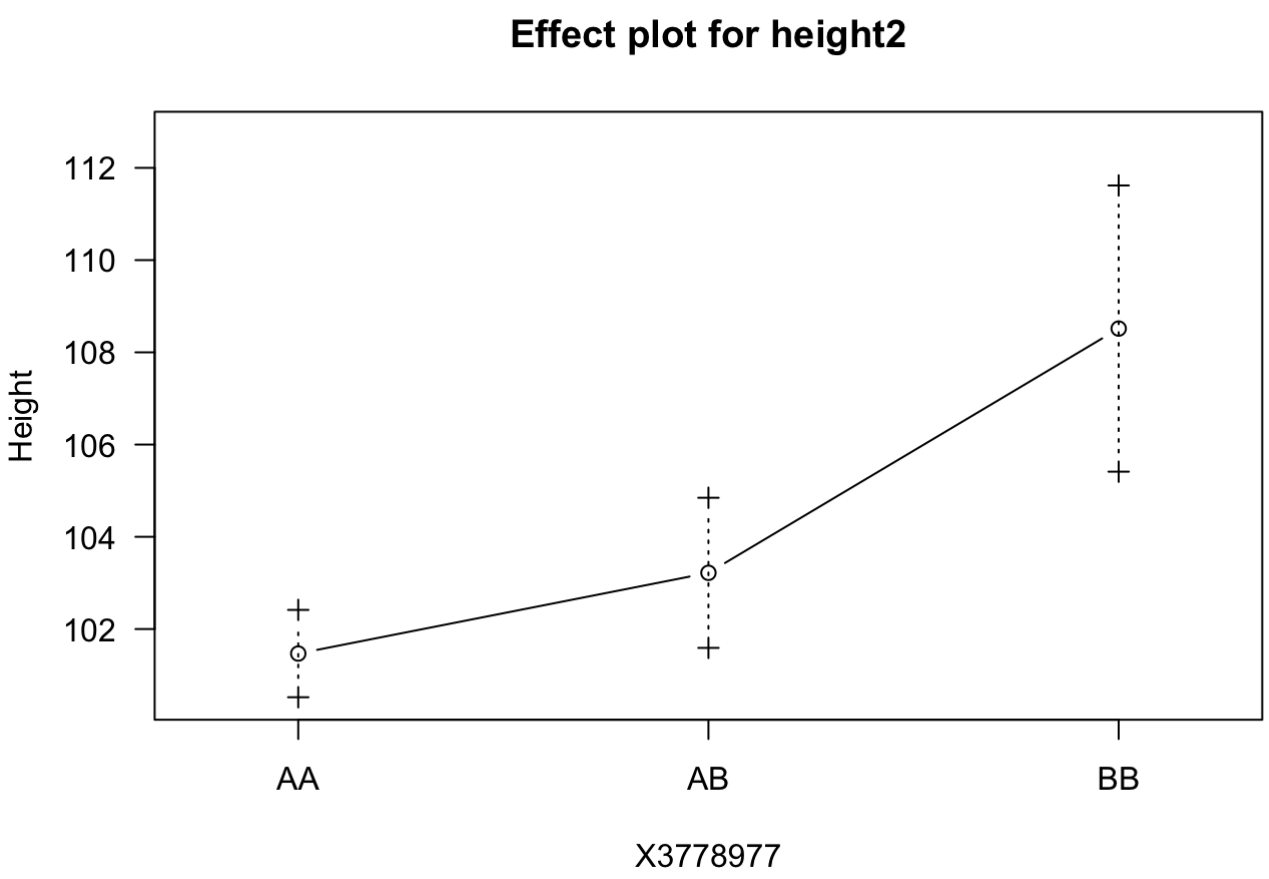
Running an effect plot for plant height

Based on your QTL information, you modify the code below to reflect the QTL data. We found two significant QTLs, one on Chromosome 2 at 3.12 cM and one on Chromosome 4 at 16.4 cM. We use **find.marker** to identify those, give them a name (height1 and height2), then use **effectplot** to see the results. Note that you have to specify the column number for the phenotype, in our case height is in Column 2. You might notice that plants with the BB genome tend to be taller! AA plants are almost 4 centimeters shorter than BB plants.

```
height1 <- find.marker(cross, chr=2, pos=3.12)
effectplot(cross, pheno.col=2, mname1=height1, main="Effect plot for height1")
```




```
height2 <- find.marker(cross, chr=4, pos=16.24)
effectplot(cross, pheno.col=2, mname1=height2, main="Effect plot for height2")
```



Now you are ready to try this on your own. It is recommended that you do a mainscan (and subsequent effect plots) for height to ensure that you get the same results. NOTE! Because some of these calculations use random number techniques such as Hidden Markov methods, your results may not be exactly the same! Then, consider modifying your code to find QTLs for “day to flow” (flow), “tillers”, and “panicles”. Note that there might not bE QTLs for one or more of these phenotypes.

A skeleton “starter code” is available to guide your coding work.