QTL Analysis using **B**ayesian **I**nterval **M**apping

Brian S. Yandell, Jee Young Moon, Samprit Banerjee, William W. Neely, Nengjun Yi

March 22, 2012

Abstract

`R/qtlbim` (`www.qtlbim.org`) provides a powerful suite of tools for model selection of the genetic architecture for traits influenced by multiple quantitative trait loci (QTL). The Markov chain Monte Carlo (MCMC) sampling approach draws samples from the more probable genetic architectures. Subsequent visualization and summary provides posterior estimates of the number and location of QTL, their main and epistatic effects, and possibly interacting covariates, or GxE interactions. This document assumes some familiarity with QTL and with Bayesian methods. Good sources are Broman (2000), Yandell et al. (2007), and Yi et al. (2005). Additional information on advances in methods can be found in Yi et al. (2007a,b).

0.1 Using `qb.scantwo`

This package provides graphical diagnostics that can help investigate "better" genetic architectures. Marginal 1-D and 2-D genome scans, analogous to `R/qtl` (`www.rqtl.org`), show the profiled contribution of QTL by locus adjusted for effects of any other QTL. Other tools identify the more probable models based on the pattern of main QTL and epistatic pairs across chromosomes. Additional diagnostic tools are briefly highlighted. This library `R/qtlbim` builds on the analytical and graphical tools in `R/qtl 1.0.8`.

Contents

1 Overview

This vignette describes the MCMC sampling routines and some of the plotting facilities available through the `R/qtlbim` package (Yandell et al. 2007). The purpose of these plots is to provide graphical tools for

1. inferring putative multiple QTL for a phenotype,
2. producing graphics and summaries of evidence for putative QTL,
3. visual diagnostics of the MCMC model selection algorithm.

Over the past year, there been numerous incremental improvements, yielding faster computation and smaller R objects. Most notably, the external directory and files created by `qb.mcmc` are now removed immediately (planned later upgrades will eliminate their need). Users with "old" style MCMC samples will be warned to upgrade using `qb.legacy`. [The old `qb.remove` is retained for compatibility, but is not needed for new `qb` objects, nor is `qb.load`.] Another important improvement is that values (results) from all routines are now self contained. The `qb` object contains the pertinent aspects of the `cross` object used to create it, and routines such as `qb.scanone` produce self-contained objects. This makes intermediate results more transportable. In addition, many objects have been made more compact, and R code efficiency has improved. Those interested in specific improvements can examine `ChangeLog.txt` in the R library area.

This document walks through the `R/qtlbim` package by demonstrating the following major functions: creation of Bayesian samples from the posterior using MCMC sampling; use of plot and summary tools to examine genetic architecture; data management in R/qtlbim. The package is invoked by the `library` command

```
> library(qtlbim)
```

1.1 Hyper data demo

This document focuses on the `hyper` dataset from `R/qtl` (Broman et al. 2003), which was initially studied in Sugiyama et al. (2001). The `hyper` dataset is stored in `R/qtl` as a `cross` object. The `R/qtlbim` package processes this `cross` object to create a `qb` object called `qbHyper`, containing the MCMC samples. The `hyper` demo shows how this is done.

```
> demo(qb.hyper.tour)
```

It is possible to directly load the already saved `qb` object with the `data` command. Following this by a call to `qb.cross` extracts a version of the `cross` object used to create the `qb` object.

```
> data(qbHyper)
> hyper <- qb.cross(qbHyper)
```

Alternatively, a `qb` object can be created by the following sequence of commands. First load the hyper data set from R/qtl, and subset on the autosomes, as `R/qtlbim` does not yet handle the X chromosome properly.

```
> data(hyper)
> hyper <- subset(hyper, chr=1:19)
```

To run the MCMC sampler on the `hyper` data we use the command

```
> hyper <- qb.genoprob(hyper, step=2)
# Now run the MCMC model selection algorithm.
# This can take several minutes.
> qbHyper <- qb.mcmc(hyper, pheno.col = 1, seed = 1616)
```

The option `seed` sets the random number seed so that this run can be repeated exactly. The `qb` object called `qbHyper` is used throughout this vignette.

1.2 Creating Bayesian interval mapping MCMC samples

This section describes in more detail how to create Markov chain Monte Carlo (MCMC) samples from the Bayesian posterior to be used for QTL mapping. The next step to mapping with the `R/qtl` package would be to use the function `calc.genoprob` to create genotype probabilities based on a Hidden Markov model. However, for Bayesian model selection, we replace `calc.genoprob` with the `R/qtlbim` function `qb.genoprob`. The function `qb.genoprob` performs some bookkeeping before calling `calc.genoprob` with the variable stepwidth option for pseudomarker positions. The probabilities for genotypes at pseudomarkers and at markers with missing data are calculated by `calc.genoprob` from the observed marker data using the multipoint method (Jiang and Zeng 1997).

The MCMC samples are created by `qb.mcmc` after running `qb.genoprob`. In the simplest case, MCMC samples are created with the following two calls:

```
> hyper <- qb.genoprob(hyper, step=2)
> qbHyper <- qb.mcmc(hyper, pheno.col = 1)
```

By default the `qb.mcmc` function prints out progress messages of the number of iterations completed. These progress messages can be suppressed by setting `verbose=FALSE`. Arguments for the routines `qb.data` and `qb.model`, described below, can be passed through `qb.mcmc`. Otherwise, default values are used. The detail below for `qb.data`, `qb.model` and `qb.mcmc` routines could be skipped in favor of default settings.

The function `qb.data` specifies the traits to be analyzed, their underlying distribution, the random and/or fixed covariates and whether to standardize or to use a boxcox transformation. Note that, the cross object can have several phenotypes and some of which could be used as covariates.

```
> qbData <- qb.data(hyper, pheno.col = 1, trait = "normal",
+     fixcov = 0, rancov = 0)
```

The `R/qtlbim` routines handle normal, binary and ordinal data. In addition, the user can specify fixed (`fixcov`) and random (`rancov`) covariate(s). [The `pheno.col`, `fixcov` and `rancov` values can be numeric indices to the phenotype names, or character strings with exact phenotype names.] Fixed covariates can be included as interacting covariates with the `intcov` option to `qb.model` (see below).

The function `qb.model` defines the model parameters, using defaults that work well in most settings. Users are probably most interested in specifying if `epistasis` is considered, the prior expected number of main effect QTLs (`main.nqtl`), and the prior expected total number of QTLs (`mean.nqtl`), which includes additional QTLs with only epistatic effects. A user may set `main.nqtl` and `mean.nqtl` based on previous QTL analysis, for example using R/qtl. Setting the maximum number of QTLs overall (`max.nqtl`) or per chromosome (`chr.nqtl`), and setting the minimum `interval` between linked QTL, can be used to restrict sampling as needed.

Typically a real data set has several traits which can be considered as covariates. The `intcov` option specifies which covariate(s) can interact with QTLs, or equivalently, which environmental factors may have GxE interactions. The `intcov` should be a vector of 0s and 1s of the same length as the `fixcov` option specified for `qb.data` (see above).

```
> qbModel <- qb.model(hyper, epistasis = TRUE, main.nqtl = 3,
+     interval = rep(5,nchr(hyper)), chr.nqtl = rep(2,nchr(hyper)),
+     depen = FALSE, prop = c(0.5, 0.1, 0.05))
```

The function `qb.mcmc` creates MCMC samples on the data and model specified. The results are initially saved in a unique directory under `mydir`, which is removed at completion of the command. Options for `qb.data` and `qb.model` can be passed directly to `qb.mcmc`, or as the objects created above.

```
> qb <- qb.mcmc(hyper, data = qbData, model = qbModel, mydir = ".",
+     n.iter = 3000, n.thin = 20, genoupdate = TRUE)
```

The `genoupdate` option simulates pseudomarker and missing marker genotypes if `TRUE`, or uses a Haley-Knott (1992) type approach if `FALSE`; the latter is faster, but not generally recommended if there are many missing genotypes or selective genotyping. `n.iter` samples are saved, thinning to one in `n.thin` from the MCMC samples to reduce serial correlation. That is, `n.iter * n.thin` samples are drawn, after an initial `n.burnin` samples (1% of total by default) are discarded to allow the chain to converge closer to the posterior distribution.

1.3 Examining a `qb` object

This package uses the S3 generic method to construct `print`, `summary` and `plot` results for routines. That is, we create an object with a call to `qb.xxx` and then plot it using the generic `plot` command, or show content summary with the generic `summary` command. The generic `print` command for most objects created with `R/qtlbim` routines invokes the generic `summary`. Manual pages show the complete set of command, print, summary and plot options.

The `qbHyper` is an object of class `qb` to which we can apply the generic `summary` or `plot` routines. We defer plots to later sections. Here we show only the summary:

```
> summary(qbHyper)

Bayesian model selection QTL mapping object qbHyper on cross object hyper
had 3000 iterations recorded at each 40 steps
with 1200 burn-in steps.
MCMC runs saved in qb object.
Trait bp ( 1 ) treated as normal .
Trait was not standardized.
Epistasis was allowed.
Prior number of QTL: 3 main, 6 total, with 13 maximum.
Minimum distance between QTL:
    1     2     3     4     5     6     7     8     9    10    11    12    13
 5.36 13.00 12.90  3.91  6.31  6.67  9.08 13.80 14.20 18.30  6.05 13.90 13.30
   14    15    16    17    18    19
17.00  5.79 10.30  4.47 11.70 18.60
Maximum number of QTL:
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
21  7  4 19 13 10  5  5  4  4 13  4  4  4 10  5 10  3  3
QTL by environment not allowed.
Interacting covariates: 0
Diagnostic summaries:
          nqtl   mean envvar varadd  varaa    var
Min.     2.000  97.42  28.07  5.112  0.000  5.112
1st Qu.  5.000 101.00  44.33 17.010  1.639 20.180
Median   7.000 101.30  48.57 20.060  4.580 25.160
Mean     6.543 101.30  48.80 20.310  5.321 25.630
3rd Qu.  8.000 101.70  53.11 23.480  7.862 30.370
Max.    13.000 103.90  74.03 51.730 34.940 65.220

Percentages for number of QTL detected:
 2  3  4  5  6  7  8  9 10 11 12 13
 2  3  9 14 21 19 17 10  4  1  0  0

Percentages for number of epistatic pairs detected:
pairs
 1  2  3  4  5  6
29 31 23 11  5  1

Percentages for common epistatic pairs:
 6:15  4:15   4:6   1:7 15:15   1:4   1:6   4:9  1:15  1:17   1:5  5:11  7:15
   63    18    10     6     6     5     4     4     3     3     3     2     2
  1:2   1:1
    2     2
```

Thus, for the 3000 MCMC samples in this object, 21% has 6 QTL (the mode), and 29% had exactly 1 epistatic pair. The most common epistatic pair, in 63% of samples with epistasis, was 6:15, or a pair of QTL on chromosomes 6 and 15.

1.4 The plot demo

The plot demo `demo(qb.plot.tour)` gives a sample of the plots available in the `R/qtlbim` package. To start the plot demo, use the command

```
> demo(qb.plot.tour)
```

For a complete set of demos, try

```
> qb.demo()
```

The plot demo begins by giving a generic plot for the `qb` object `qbHyper`. The `R/qtlbim` generic `qb` plot is analogous to the generic `R` plot for linear model objects. Where the generic plot for a linear model object shows a sequence of graphics whose purpose is to aid in the initial results of model fitting, the generic plot function for `qb` objects shows a sequence of graphics whose purpose is to give an initial assessment of the results produced by the MCMC algortihm. The generic plot for the `qb` object `qbHyper` created above is shown with the command

```
> plot(qbHyper)
```

The generic plot function shows a sequence of plots that include time series plots of the mcmc chain, jittered plots of QTL by chromosome and others. The sequence of plots appearing in the plot demo is listed below. The actual plots are shown later in this document under the section Useful Plots.

The list of plots shown by the generic plot function.

1. A time series plot of the mcmc chain runs. This is shown in Figure 4, where it was created by the command `plot(qb.coda(qbHyper))`.
2. A jittered plot of QTL by chromosome. This plot, produced separately by `plot(qb.loci(qbHyper))`, can be seen in Figure 5 for two chromosomes.
3. A model selection plot by chromosome. This plot is identical to `plot(qb.BayesFactor(qbHyper))` shown in Figure 6.
4. Plot of QTL posterior for loci plus smooth estimates of QTL effects. This plot is the same as the plot generated by `plot(qb.hpdone(qbHyper))`. Figure 7 shows the result of this command.
5. A plot of epistatic effects if such effects are allowed. Figure 8 shows the result of the command `plot(qb.epistasis(qbHyper))`.
6. Summary diagnostics as histograms and boxplots by number of QTL. This final diagnostic plot can be generated separately by the command `plot(qb.diag(qbHyper))`. Figure 9 shows the result of this command.

2 Marginal 1-D and 2-D Genome Scans

This document describes 1-D and 2-D Bayesian genome scan routines available in the `R/qtlbim` package. In the present context, the term "scan" refers to methods based on constructing one or two dimensional profiles of QTL likelihoods or posterior distributions. These new scan routines in `R/qtlbim` are analogous to the routines `scanone` and `scantwo` from the `R/qtl` package. On a practical level, using `R/qtlbim` scan routines is very similar to using `R/qtl`'s `scanone` and `scantwo` methods. The key difference between the scan routines in `R/qtlbim` and the scan routines in `R/qtl` lies in the technique used for constructing QTL summaries. `R/qtlbim` extends `R/qtl` by providing the ability to generate Markov chain Monte Carlo (MCMC) samples from a posterior distribution for the genetic architecture of a trait. Furthermore the putative genetic architectures sampled can include an arbirary number of QTL.

2.1 Using `qb.scanone`

The `R/qtlbim` package's scan routines are called `qb.scanone` and `qb.scantwo`. Because these scans are motivated by Bayesian MCMC techniques we refer to `qb.scanone` and `qb.scantwo` collectively as "qb.scans" or "qb.scan routines". The utility of the qb.scan routines lies in their ability to provide interpretable summaries of the high-dimensional MCMC samples. The scan summaries use ideas of

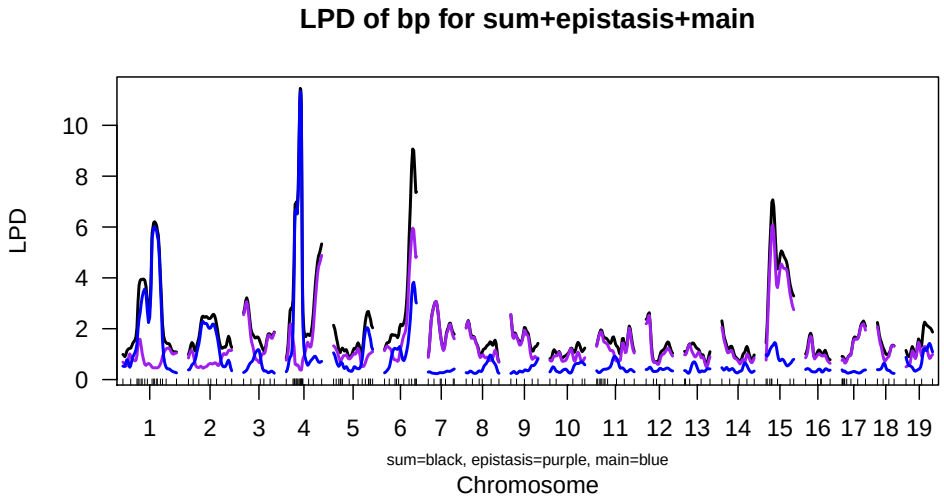


Figure 1: Plot of `qb.scanone` for LPD of hyper data. Notice the posterior concentrated on chromosomes 1, 4, 6 and 15.

Bayesian model averaging to explore the most probable models given the data. For example, in a one dimensional genome scan, we might consider the contribution of each potential locus averaging over all sampled models that include that locus. This allows us to adjust for the possible effects of all other loci by examining the marginal distributions. This has the advantage of reducing variation explained by other loci and reducing bias due to linked loci. Thus a one dimensional marginal scan can be informative about higher-order models directly without bias or variance inflation. Although the development of the `qb.scan` routines is motivated by Bayesian techniques, the interpretation of qb.scans involve a mix of frequentist and Bayesian ideas. In what follows we show the resolving power of low-dimensional scans that condition on the presence of other QTL using simulated data with one QTL and the `hyper` data set.

This section illustrates the basic uses and interpretation of the qb.scan routines using simulated data and the `hyper` data. The object `qbHyper` created above contains the results of the MCMC run. Each iteration of the Monte Carlo chain represents a single QTL model. The entire Monte Carlo chain represents a sample from the posterior distribution of all possible models. One simple summary of the MCMC sample is the LPD profile, or the Log Posterior Density for a QTL at each locus. The LPD is analogous to the classical LOD, or Log ODds. A single QTL LPD can be computed with `R/qtl`'s `scanone` using `method="im"`. The marginal LPD from `qb.scanone`, however, provides the contribution to LPD of a QTL at a locus *adjusting* for all other possible QTL. [For a technical interpretation, see the section on Theoretical Development.] A summary and plot of the LPD is carried out as follows.

```
> temp <- qb.scanone(qbHyper,type="LPD")

> plot(temp)

> summary(temp)

LPD of bp for main,epistasis,sum

    n.qtl   pos m.pos e.pos   main epistasis    sum
1  1.3310 67.80 67.80 67.80  5.972     0.459  6.172
2  0.3477 51.90 51.90 42.63  2.011     0.492  2.396
3  0.1453 30.63 30.63  8.76  1.145     3.068  1.678
4  1.3770 29.50 29.50 29.50 11.329     0.377 11.453
5  0.2447 68.87 68.87 82.00  2.029     1.095  2.525
6  0.8383 59.00 59.00 59.00  3.745     5.959  9.069
7  0.1553 15.28 55.60 15.28  0.418     3.029  3.042
8  0.1320 56.93 59.00 17.52  0.946     1.626  1.488
```

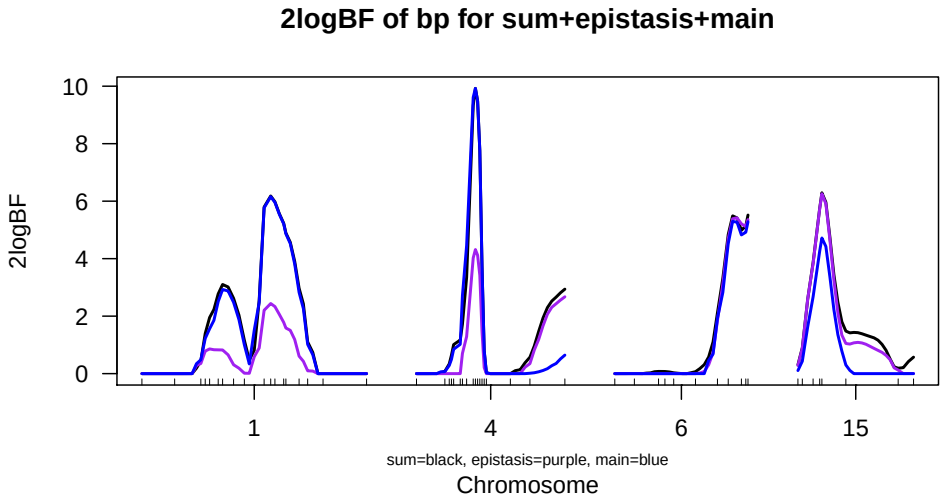


Figure 2: The `qb.scanone` results for the `hyper` data restricted to chomosomes 1,4,6 and 15.

```
9  0.1173 12.00 64.87 12.00  0.662     2.561  2.548
10 0.0947 37.95 75.40 37.95  0.581     0.840  0.984
11 0.1717 17.50 39.57 13.10  0.916     1.831  1.644
12 0.0947  1.10 46.55  1.10  0.452     2.197  2.368
13 0.0767 24.40 28.40 14.23  0.648     1.346  1.432
14 0.0840  0.00 46.35  0.00  0.621     2.059  2.310
15 0.9607 17.50 17.50 17.50  1.309     6.019  6.977
16 0.0813  8.37  8.37 10.46  0.396     1.710  1.744
17 0.1123 50.30  1.10 50.30  0.383     1.943  2.090
18 0.0663  2.20 14.20  2.20  0.599     2.070  2.245
19 0.1117 55.70 53.62 55.70  1.211     0.985  1.869
```

Figure 1 shows the LPD concentrated on chromosomes 1, 4, 6 and 15, which is consistent with other findings for these data (Sugiyama et al. 2001). The blue lines in the plot indicate main effects, the purple indicate epistatic effects and black curves (where visible) represent the sum of main and epistatic effects.

Figure 2 shows 2log(BF), or twice the log of the Bayes factor, measuring the strength of evidence (> 2.1 is high) for a QTL. In order to examine the effects on 1, 4, 6 and 15 more closely, we can plot subsets of chromosomes by using the plot command `plot(temp, chr=c(1,4,6,15))`.

```
> temp <- qb.scanone(qbHyper, type = "2logBF")
```

```
> plot(temp,chr=c(1,4,6,15))
```

2.2 Using `qb.scantwo`

The function `qb.scantwo` gives a two dimensional scan that allows us to look for possible epistatic effects between putative QTL. To run `qb.scantwo` on the `hyper` data set, we again use the MCMC samples. The summary and the plot in Figure 3 shows strong evidence for the 6:15 epistasis, and good evidence for a 4:15 epistatic interaction that was missed in earlier analyses.

```
> temp <- qb.scantwo(qbHyper, chr = c(4, 6, 15))
> summary(temp, digits = 2)
```

```
upper: heritability of bp for epistasis
lower: heritability of bp for full

        n.qtl l.pos1 l.pos2 lower u.pos1 u.pos2 upper
```

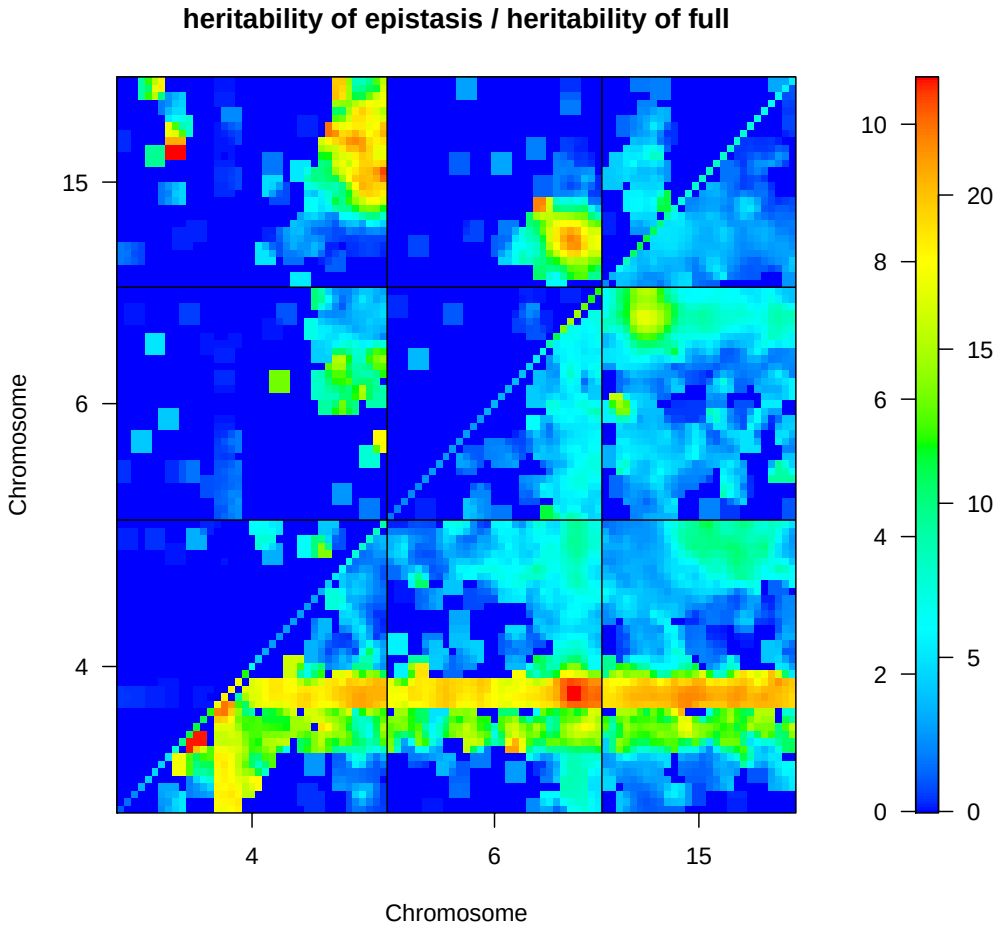


Figure 3: A plot of a `qb.scantwo` scan of the `hyper` data showing results for chromosomes 1, 4,6, and 15. Note the main effect from the QTL on chromosome 4 and the epistatic effect between the pairs of QTLs on chromosomes 4 and 15 and 6 and 15.

```
c4 :c4  0.417   16.4   18.6  23.6  54.35   65.6  6.22
c4 :c6  1.185   29.5   59.0  23.8  72.11   21.9  8.08
c4 :c15 1.452   28.4   31.5  21.7  14.20   41.6 10.67
c6 :c6  0.111   59.0   59.0  14.6   7.35   49.2  1.55
c6 :c15 1.004   59.0   19.5  16.9  59.00   19.5  9.77
c15:c15 0.261   19.5   19.5  11.5  23.50   27.5  5.04
```

```
> plot(temp)
```

Using the results from the two-dimensional qb.scans of the simple simulated data as a guide, the plot of `qb.scantwo` shows a main effect from a QTL on chromosome 4 and epistatic effects between the pairs of QTLs on chromosomes 4 and 15 and 6 and 15.

2.3 Types of Scan Summaries

We have created several types of scan summaries, illustrated below. These include the following LPD, heritability, variance components, parameter estimates, cell means, posterior probabilities and Bayes factors. Below we detail what these are and how they are calculated.

For each type, we can provide a summary scan, and in addition provide detail broken down by main effects, epistatic effects, and/or GxE (genotype by environment, or genotype by covariate) interactions.

These breakdowns can be further divided into Cockerham (1954; see Kao and Zeng 2002) type effects (additive and dominance for main effects, or the four epistatic interactions of aa, ad, da, dd) if desired.

- `count` gives the count of the number of MCMC samples including this locus. Currently this can be viewed on a log scale using type `log10`.
- `posterior` is the Bayesian posterior probability, basically the `count` divided by the total number of MCMC samples.
- `BF` provides the Bayes factor comparing the model with and without this locus. It is more easily viewed as `2logBF`.
- `estimate` gives model parameter estimates for main effects, epistasis, and GxE interactions.
- `cellmean` provides marginal means at a locus, adjusted for all other model effects from other QTL and covariates.
- `variance` yields the variance components for QTL effects associated with a particular locus.
- `heritability` is actually at this point explained variation. In a future release we may distinguish Rsquared and idealized heritability.
- `LPD` is the log posterior density, adapted from Morton's (1995) log odds ratio (LOD) used in human genetics to LOD maps by Lander and Botstein (1989). The LPD for QTLs was introduced by Sen and Churchill (2001). It tests presence or absence of a QTL at a locus, adjusting for all other possible model effects (other QTL, epistasis and GxE). The `LPD`, the `LR` or likelihood ratio, and the `deviance` are detailed in the next section.
- `detection` is the posterior probability of detection of a QTL at a locus.

3 Model Selection for Genetic Architecture

The `R/qtlbim` model selection tools do the following:

1. evaluate Bayes factor for number or chromosome pattern of QTL (`qb.bf`);
2. examine proximity of sampled architectures (`qb.best`);
3. measure closeness of sampled architectures to target (`qb.close`).
4. one-dimensional (`qb.scanone`) or two-dimensional (`qb.scantwo`) genome scan;
5. characterize genetic architecture (`qb.arch`);
6. stepwise regression on genetic architecture (`step.fitqtl`);

In addition, several new routines begin to examine linked QTL:

1. examine multiple loci (`qb.multloci`);
2. find main and epistatic modes (`qb.mainmodes`, `qb.epimodes`);
3. split chromosomes for linked QTL (`qb.split.chr`);

3.1 What is the Best Model?

It is well and good to be able to explore possible genetic architectures, but what is the best? Here we start by defining the best genetic architecture as the most probable combinations of QTLs across chromosomes and any epistatic pairs given the data. Formally, this is the pattern of QTL with the highest posterior probability. In fact, this document focuses on assessing the chromosome pattern of QTLs. The approach has been found to be comparable in power to stepwise regression approaches (Manichaikul et al. 2008).

The routine `qb.bf` (or `qb.BayesFactor`) can compute the posterior and Bayes factor for the more probable patterns.

```
> bf <- qb.bf(qbHyper, item = "pattern")
> summary(bf)

$pattern
                  nqtl posterior    prior    bf   bfse
1,4,4,6,15,6:15      6   0.00300 3.15e-07 75.30 25.100
1,1,4,5,6,15,6:15    7   0.00267 2.97e-07 71.00 25.100
1,1,4,6,15,6:15      6   0.00600 8.68e-07 54.70 12.800
```

```
1,2,4,6,15,6:15      6   0.00767 1.20e-06 50.30 10.500
1,4,6,15,6:15        5   0.03400 5.86e-06 45.80  4.460
1,4,6,6,15,6:15      6   0.00467 8.52e-07 43.30 11.500
1,2,4,5,6,15,6:15    7   0.00267 5.18e-07 40.70 14.400
1,4,5,6,15,6:15      6   0.00500 1.73e-06 22.80  5.880
1,4,6,15,15,6:15     6   0.00300 1.05e-06 22.50  7.490
1,1,2,4              4   0.00300 3.43e-06  6.92  2.300
1,2,4                3   0.00733 2.57e-05  2.26  0.479
1,1,4                3   0.00400 1.51e-05  2.09  0.603
1,4,19               3   0.00300 1.45e-05  1.63  0.543
1,4                  2   0.01430 1.13e-04  1.00  0.151
```

The pattern with the highest posterior probability is 1,4,6,15,6:15, whereas the pattern with highest Bayes factor is 1,4,4,6,15,6:15. Patterns are represented a chromosome identifiers separated by commas; epistatic pairs of chromosomes are joined by a colon. The `qb.bf` summary model-averages over all possible loci on each chromosome. That is, with MCMC sampling, we find the frequency of the chromosome pattern while ignoring the actual loci values.

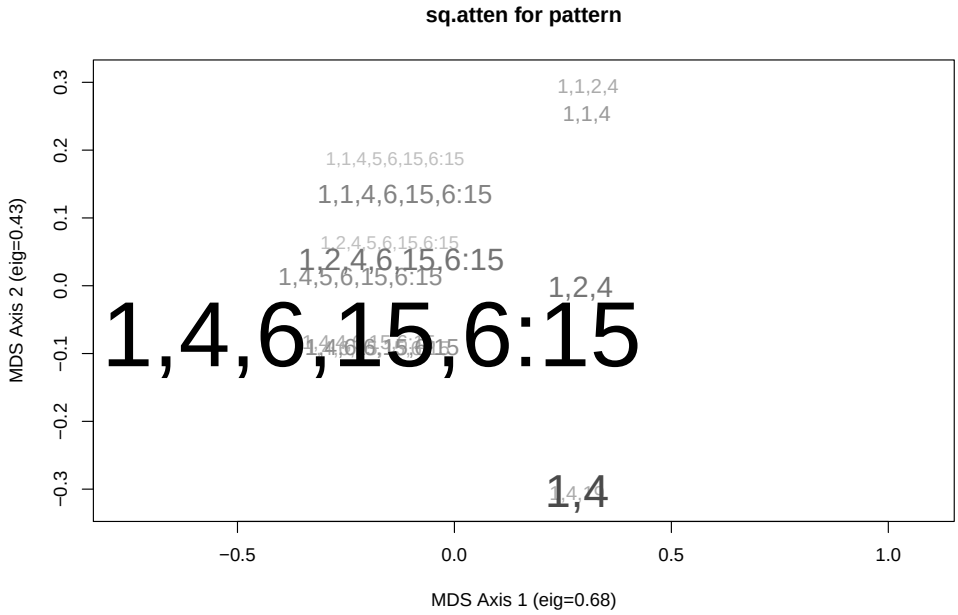
This might be enough. However, we can now ask for the most probable chromosome pattern, what are the best estimates of loci? These are the averages of loci positions for those models that include exactly these chromosome patterns. The routine `qb.best` (or `qb.BestPattern`) can perform this task, and a few more.

```
> best <- qb.best(qbHyper)
> summary(best)
```

```
Best pattern(s) by sq.atten score
    n.qtl chrom locus locus.LCL locus.UCL variance variance.LCL variance.UCL
247 0.803     1  69.9    24.449    95.799     4.33       0.0345         9.87
245 0.880     4  29.5    14.200    74.300     9.10       0.0885        17.20
248 0.710     6  59.0    13.833    66.700     4.73       0.1300        10.50
246 0.845    15  19.5    13.100    55.700     2.64       0.0823         7.31

Summary by better patterns
                  terms   percent    score cluster
1,4,6,15,6:15         4 3.4000000 4.000000       1
1,2,4,5,6,15,6:15     6 0.2666667 3.956954       1
1,4,4,6,15,6:15       5 0.3000000 3.956954       1
1,1,4,6,15,6:15       5 0.6000000 3.923116       1
1,4,5,6,15,6:15       5 0.5000000 3.919431       1
1,2,4,6,15,6:15       5 0.7666667 3.876550       1
1,1,4,5,6,15,6:15     6 0.2666667 3.842548       1
1,4,6,6,15,6:15       5 0.4666667 3.822012       1
1,4,6,15,15,6:15      5 0.3000000 3.809098       1
1,4                   2 1.4333333 2.000000       2
1,2,4                 3 0.7333333 2.000000       2
1,4,19                3 0.3000000 2.000000       2
1,1,4                 3 0.4000000 1.919431       3
1,1,2,4               4 0.3000000 1.919431       3

Maximum number of QTL in architecture: 11
```

The best pattern is by design the most probable, but we now have estimates of the `locus` and `variance` contribution for each QTL. We can view more pattern details, say the top 3 patterns, with the option `n.best = 3`. We can see how this pattern compares to other patterns in a few plots.

```
> plot(best)
```

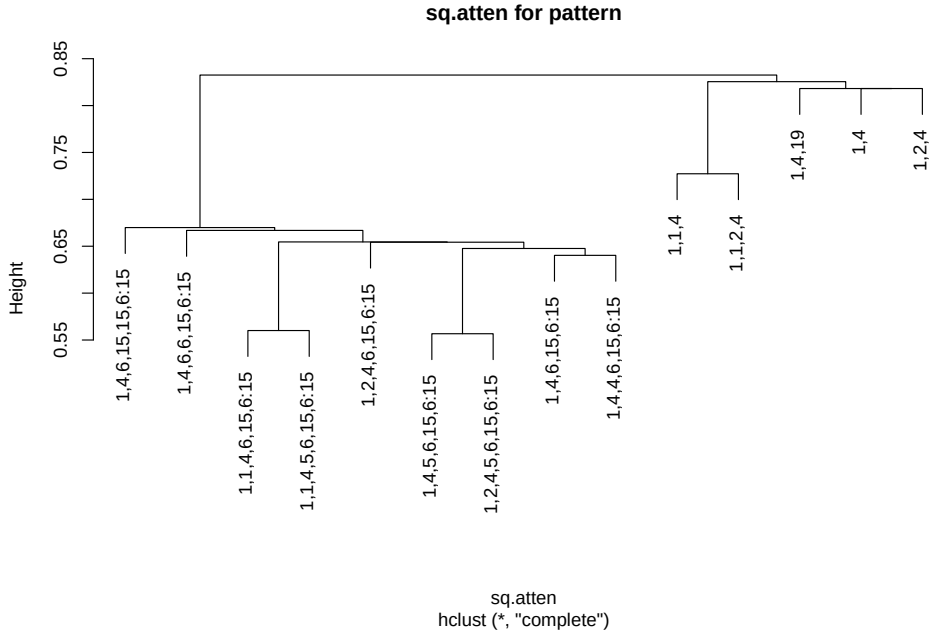


The font size of a pattern is determined by its posterior probability. The 2-D multidimensional scaling (MDS) projection is based on the `score.type` (see below). Notice that models that overlap with 1,4,6,15,6:15 are plotted near that pattern. Other patterns with little overlay are some distance away.

The default `score.type` is `sq.atten`, the square of the attenuation. When comparing two models, consider a QTL locus estimated by each to be on the same chromosome. The attenuation is $(1 - 2r)$, with r the genetic distance (in Morgans) between the estimates. If the loci agree exactly, there is no attenuation ($r = 0$). Loci on different chromosomes for different models have a score contribution of 0. The scores are added up, trying in the process to match of QTL as best as possible between any two genetic architectures. Other `score.type`s are `attenuation` (signed or not), `recombination`, `distance`, and explained `variance`. The latter provides a one-dimensional ordering of models based on overall fit.

It is possible to examine the patterns in another way, by plotting a dendrogram based on hierarchical clustering.

```
> plot(best, type = "hclust")
```



The default for method of model averaging of the `locus` and `variance` for `qb.best` is to average over loci from all MCMC samples that include a particular pattern–that is, average over all patterns that have the target `nested` within them. Instead, we can model average over all MCMC samples, or only those with an exact match to the best pattern. The `all` average uses the most MCMC samples per locus, while the `exact` typically involves very few samples, those that exactly match a particular pattern. There is a tradeoff of bias and variance in the choice of these methods, although bias appears empirically to be small due to the way MCMC samples cluster around more probable loci. Below are the three choices for inclusion in model averaging. It is also possible to change the way the `center` is determined (default is `"median"`, but `"mean"` is an alternative). The plots and summaries (not shown) change slightly as well, as all better patterns are altered similarly.

```
> qb.best(qbHyper, include = "all")$model[[1]]
```

```
        n.qtl chrom locus locus.LCL locus.UCL variance variance.LCL
247 1.3310000     1  69.9  24.06667  96.18000 4.291848   0.03516970
245 1.3770000     4  29.5  14.20000  74.30000 9.206616   0.08047250
248 0.8383333     6  59.0   9.80000  66.70000 4.065665   0.04463393
246 0.9606667    15  19.5  13.10000  58.26667 2.442734   0.04279294
    variance.UCL
247    10.027673
245    17.222186
248    10.274912
246     7.205367
```

```
> qb.best(qbHyper, include = "nested")$model[[1]]
```

```
        n.qtl chrom locus locus.LCL locus.UCL variance variance.LCL
247 0.8026667     1  69.9  24.44875   95.7985 4.331837   0.03452814
245 0.8800000     4  29.5  14.20000   74.3000 9.098802   0.08845976
248 0.7096667     6  59.0  13.83333   66.7000 4.725800   0.12963260
246 0.8450000    15  19.5  13.10000   55.7000 2.638343   0.08227567
    variance.UCL
247     9.871876
245    17.239369
248    10.517350
246     7.310082
```

```
> qb.best(qbHyper, include = "exact")$model[[1]]
```

```
    n.qtl chrom locus locus.LCL locus.UCL  variance variance.LCL variance.UCL
247 0.034     1  69.9      43.7     77.60  4.768429     1.897535    10.746897
245 0.034     4  29.5      29.5     30.60 11.538096     5.841942    17.412872
248 0.034     6  61.2      54.1     66.70  5.173255     1.391078    10.752676
246 0.034    15  17.5      13.1     26.45  3.183654     1.162633     7.181975
```

3.2 How Close are Other Models to a Target?

A target model might arise from another study, or from another analysis of the same dataset. Right here, we will use the most probably model as target, but the target object is simply a data frame with columns for `chrom`, `locus` and `variance`. [If `variance` is omitted, it is filled in with 0s.] Here is the target we are using:

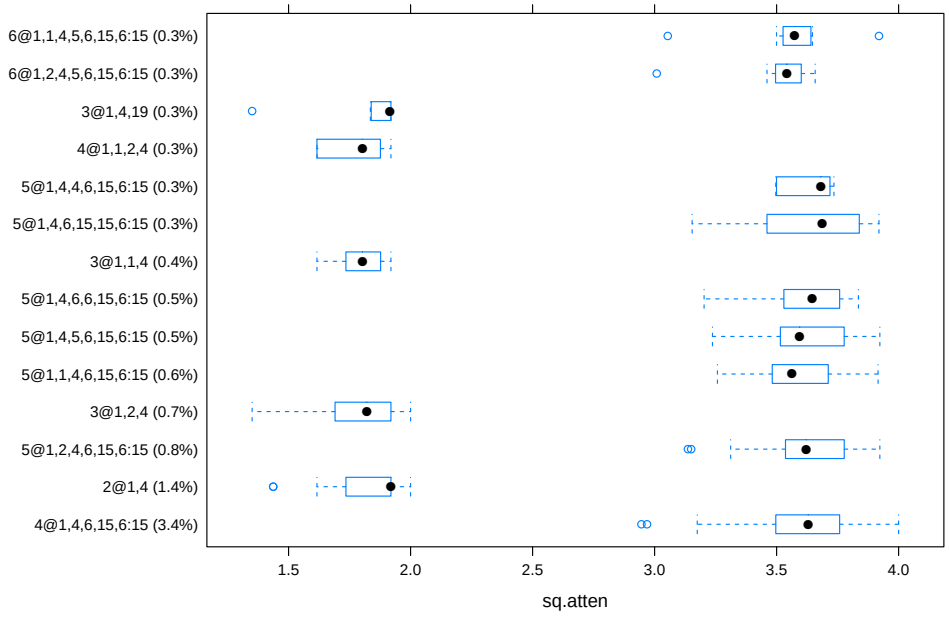
```
> target <- best$model[[1]]
```

The routine `qb.close` gives a score comparison for each MCMC realization. These are summarized over chromosome pattern, or over number of QTL using boxplots.

```
> close <- qb.close(qbHyper, target)
> summary(close)

 target for score sq.atten
    chrom locus variance
247     1  69.9 4.331837
245     4  29.5 9.098802
248     6  59.0 4.725800
246    15  19.5 2.638343

 score by sample number of qtl
    Min. 1st Qu. Median  Mean 3rd Qu.  Max.
2  1.437   1.735  1.919 1.834   1.919 2.000
3  1.351   1.735  1.916 1.900   1.919 2.916
4  1.270   1.916  2.437 2.648   3.574 4.000
5  1.295   1.919  2.835 2.798   3.611 4.000
6  1.257   2.254  3.451 3.029   3.648 4.000
7  1.351   2.836  3.492 3.212   3.677 3.923
8  1.329   3.237  3.574 3.340   3.744 4.000
9  1.295   3.272  3.576 3.334   3.727 4.000
10 2.000   3.432  3.614 3.475   3.762 4.000
11 1.899   3.382  3.525 3.428   3.697 3.923
12 1.391   2.702  3.574 3.174   3.661 3.759
13 3.694   3.694  3.694 3.694   3.694 3.694

 score by sample chromosome pattern
                    Percent  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
4@1,4,6,15,6:15       3.400 2.946   3.500  3.630 3.613   3.758 4.000
2@1,4                 1.430 1.437   1.735  1.919 1.832   1.919 2.000
5@1,2,4,6,15,6:15     0.767 3.137   3.536  3.622 3.611   3.777 3.923
3@1,2,4               0.733 1.351   1.700  1.821 1.808   1.919 2.000
5@1,1,4,6,15,6:15     0.600 3.257   3.484  3.563 3.575   3.698 3.916
5@1,4,5,6,15,6:15     0.500 3.237   3.515  3.595 3.622   3.777 3.923
5@1,4,6,6,15,6:15     0.467 3.203   3.541  3.646 3.631   3.757 3.835
3@1,1,4               0.400 1.616   1.735  1.803 1.790   1.858 1.919
5@1,4,6,15,15,6:15    0.300 3.154   3.461  3.687 3.642   3.839 3.919
5@1,4,4,6,15,6:15     0.300 3.497   3.500  3.681 3.630   3.719 3.735
4@1,1,2,4             0.300 1.616   1.616  1.803 1.775   1.876 1.919
3@1,4,19              0.300 1.351   1.839  1.916 1.837   1.919 1.919
6@1,2,4,5,6,15,6:15   0.267 3.009   3.513  3.542 3.493   3.584 3.658
6@1,1,4,5,6,15,6:15   0.267 3.054   3.540  3.574 3.557   3.638 3.919
```

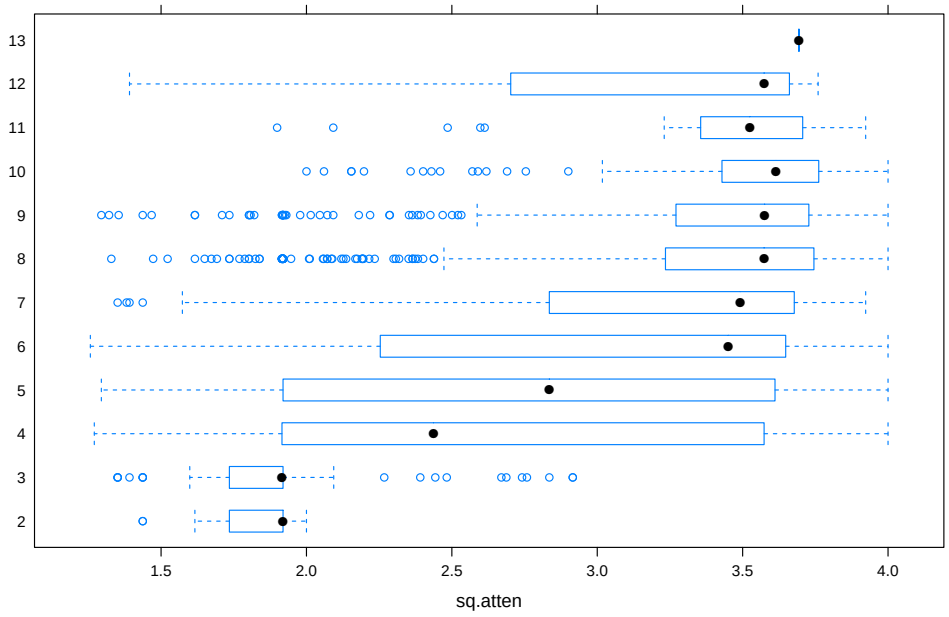
It is more intuitive to look at the boxplots. Notice how patterns that miss the 6:15 interaction have much lower attenuation scores.

```
> plot(close)
```



Now examine close-ness summarized by number of QTL in the sample. Notice that the samples with 6 or more QTL essentially pick up the four target QTL. It is common for Bayesian interval mapping to "overfit". This is not necessarily a bad thing. Some of the QTL will have small effects. Other tools such as `qb.scanone` can be used to investigate which QTL fit have weak evidence.

```
> plot(close, category = "nqtl")
```



3.3 ANOVA confirmation

The routines `qb.arch` and `step.fitqtl` can be helpful to refine model selection for genetic architecture. They are illustrated in the document on a prototype QTL study of the hyper dataset (see Summary).

Here we start with the `best` model and use `step.fitqtl` to step-by-step reduce the model to key main effects and interactions, preserving hierarchy. The following uses `R/qtl` tools `calc.genoprob`, `sim.geno` and `makeqtl`, plus `R/qtlbim`'s `step.fitqtl`, which calls `fitqtl` multiple times.

```
> hyper.arch <- qb.arch(best)
> hyper.arch

main QTL loci:
    [,1]   [,2]   [,3]   [,4]
chr "1"    "4"    "6"    "15"
pos "69.9" "29.5" "59.0" "19.5"

Epistatic pairs by qtl, chr, pos:
       qtla qtlb chra chrb posa posb
pair 1    3    4    6   15   59 19.5
Epistatic chromosomes by connected sets:
15,6

> hyper.sub <- subset(hyper, chr = hyper.arch$qtl$chr)
> n.draws <- 8
> hyper.sub <- sim.geno(hyper.sub, n.draws = n.draws, step = 2,
+     error = 0.01)
> qtl <- makeqtl(hyper.sub, as.character(hyper.arch$qtl$chr), hyper.arch$qtl$pos)
```

Now we run stepwise backward elimination, preserving hierarchy. The `step.fitqtl` routine is simply a wrapper for `R/qtl`'s `fitqtl` using an analogy to `R`'s `step` function.

```
> hyper.step <- step.fitqtl(hyper.sub, qtl, pheno.col = 1, hyper.arch)

> summary(hyper.step$fit)

Full model result
----------------------------------

       df        SS         MS     LOD     %var Pvalue(Chi2) Pvalue(F)
Model   5  5312.444 1062.48884 19.4144 30.06658            0         0
Error 244 12356.492   50.64136
Total 249 17668.936


Drop one QTL at a time ANOVA table:
----------------------------------
               df Type III SS    LOD   %var F value Pvalue(F)
1@69.9          1      1206.6  5.058  6.829   23.83  1.91e-06 ***
4@29.5          1      2678.4 10.651 15.159   52.89  4.78e-12 ***
6@58.0          2      1251.8  5.239  7.085   12.36  7.71e-06 ***
15@19.5         2      1135.6  4.773  6.427   11.21  2.20e-05 ***
6@58.0:15@19.5  1       850.9  3.615  4.816   16.80  5.65e-05 ***
```

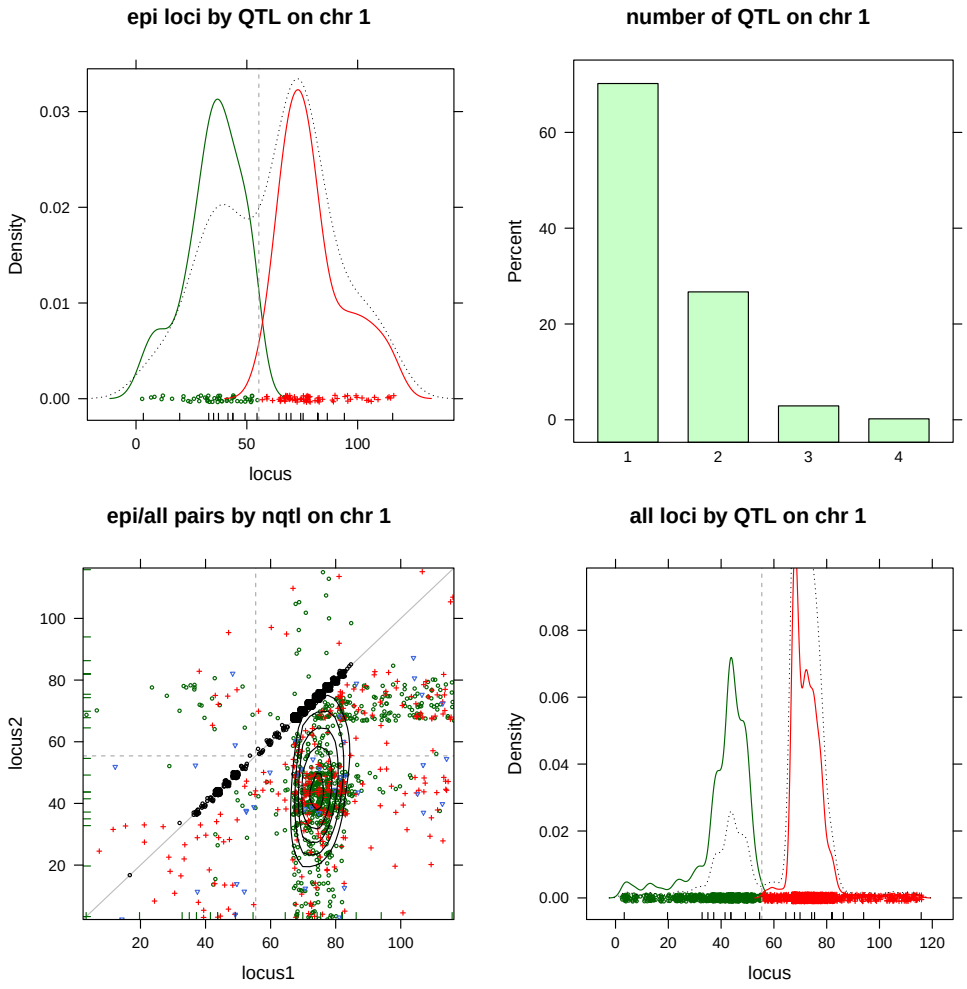
In this case, there was no reduction in the selected model, meaning the four main QTL and the 6:15 epistatic pair are confirmed. There are other exciting new model selection approaches in recent releases of `R/qtl`. See also Manichaikul et al. (2008) for empirical comparison of methods.

3.4 Multiple Linked Loci

Sometimes there appear to be evidence for linked loci. While 2-dimensional scans with `scantwo` or `qb.scantwo` can disambiguate such situations, it can be helpful to have tools to look finer, and even to break chromosomes apart.

The routine `qb.multloci` allows a look at evidence for two or more linked QTL. The upper right panel shows the posterior for number of linked QTL. The lower right panel shows the density broken up by a reasonable guess at the number of QTL (the highest value with at least 20% of the samples). The suggested break is based on the valley between peaks, using discriminant analysis. The upper left panel shows the epistatic pairs, and the lower left panel shows a two way plot of singletons (diagonal), pairs, triplets (as three pairs), etc.

```
> mult <- qb.multloci(qbHyper, chr = 1)
> plot(mult)
```



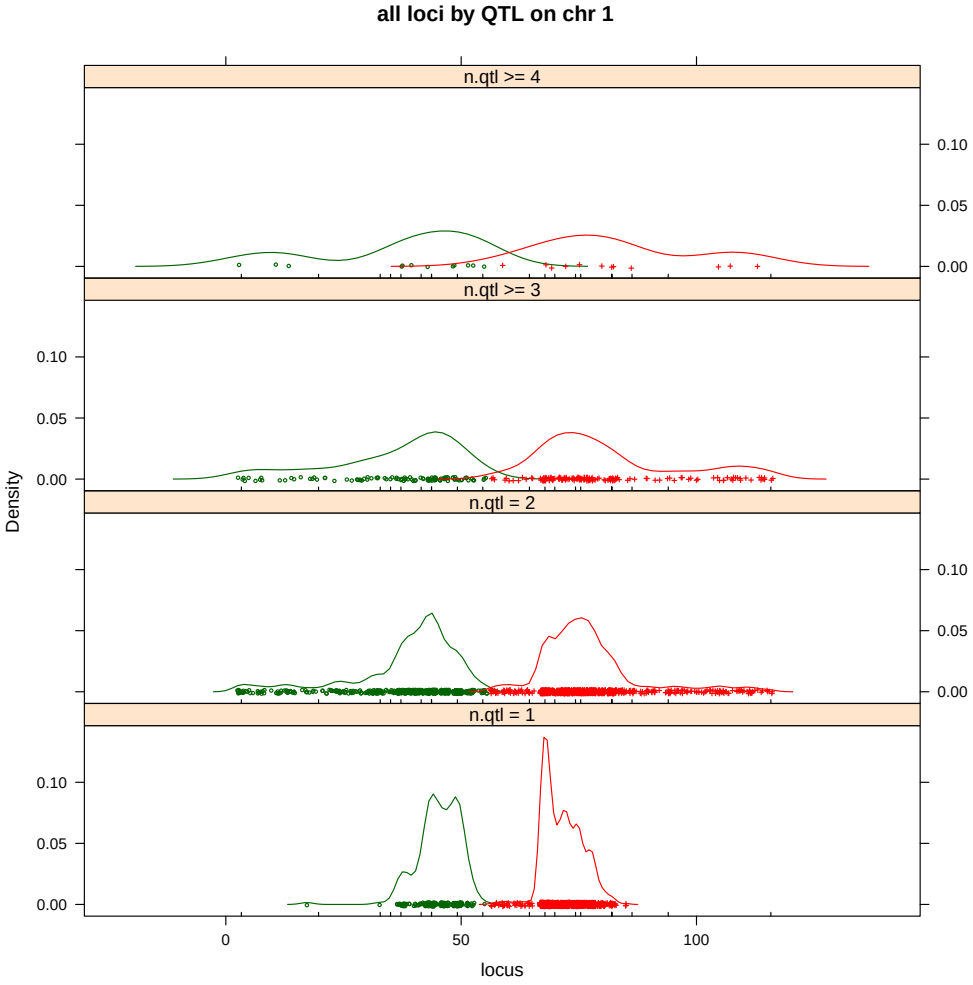

```
> summary(mult)
Posterior Percent by Number of QTL

   1    2    3    4
70.2 26.7  2.9  0.2
Estimated Number of QTL: 2

Peaks
       1        2
43.76686 68.11157
Valleys
       1
55.41529
QTL Summaries
       Min. 1st Qu. Median  Mean 3rd Qu.  Max.   Pct. Ties
QTL 1  3.30    37.2   43.7 39.63   46.45  54.6  30.77 1.53
QTL 2 57.08    67.8   72.1 73.73   77.60 115.8 102.33 8.63
```

It is helpful sometimes to separate out samples with different number of QTL. This can be done with the `merge` option.

```
> summary(mult, merge = FALSE)
```

```
Posterior Percent by Number of QTL

   1    2    3    4
70.2 26.7  2.9  0.2
Estimated Number of QTL: 2

Peaks
       1        2
43.76686 68.11157
Valleys
       1
55.41529
QTL Summaries
$`nqtl = 1`
       Min. 1st Qu. Median  Mean 3rd Qu.  Max.  Pct. Ties
QTL 1 17.65    43.7  46.45 45.31    49.2 54.60  6.33    0
QTL 2 57.08    67.8  72.10 71.54    74.3 84.15 63.87    0

$`nqtl = 2`
       Min. 1st Qu. Median  Mean 3rd Qu.  Max.  Pct. Ties
QTL 1  3.30    37.2   41.5 38.55   46.45  54.6 20.37 0.33
QTL 2 57.08    72.1   75.4 76.78   79.80 115.8 33.03 6.67

$`nqtl >= 3`
       Min. 1st Qu. Median  Mean 3rd Qu.  Max. Pct. Ties
QTL 1  3.30   28.43  40.43 36.12   46.45  54.6 3.67 1.07
QTL 2 57.08   69.90  77.60 80.87   86.30 115.8 5.03 1.83

$`nqtl >= 4`
       Min. 1st Qu. Median  Mean 3rd Qu.  Max. Pct. Ties
QTL 1  3.30   31.29  41.53 36.88   49.88  54.6  0.4 0.13
QTL 2 59.55   71.55  80.90 83.37   90.95 113.6  0.4 0.13
```

```
> plot(mult, merge = FALSE)
```



The peaks and valleys are computed with `qb.mainmodes`. While this routine is visible to the user, it is seldom actually needed. `qb.epimodes` serves a similar function for epistatic pairs only.

Once a logical split for a chromosome has been established, we can use `qb.split.chr` to formalize the split. By default, it uses the results from `qb.mainmodes`.

```
> qbHyper <- qb.split.chr(qbHyper)
> qb.get(qbHyper, "split.chr")
```

```
$`1`
       1
55.41529

$`4`
       1
46.21198
```

The split can be negated by the argument `split = NULL`. A few routines now use this split, and more are planned. For now, `qb.scanone`, `qb.scantwo` and `qb.bf` take advantage of this. Chromosomes are recoded as chr.1, chr.2, etc.

```
> qb.bf(qbHyper, item = "pattern")
```

```
$pattern
                     nqtl posterior    prior    bf   bfse
1.1,1.2,4.1,6,15,6:15    6   0.00533 8.49e-07 52.10 13.000
1.2,4.1,6,15,6:15        5   0.03170 5.54e-06 47.30  4.780
1.2,2,4.1,6,15,6:15      6   0.00700 1.26e-06 45.90  9.980
```

```
1.2,4.1,6,6,15,6:15      6   0.00433 9.03e-07 39.80 11.000
1.2,4.1,5,6,15,6:15      6   0.00467 1.82e-06 21.20  5.670
1.2,4.1,6,15,15,6:15     6   0.00267 1.16e-06 19.00  6.720
1.2,2,4.1                3   0.00700 2.57e-05  2.26  0.491
1.1,1.2,4.1              3   0.00333 1.51e-05  1.83  0.577
1.2,4.1,19               3   0.00267 1.45e-05  1.52  0.537
1.2,4.1                  2   0.01370 1.13e-04  1.00  0.155
```

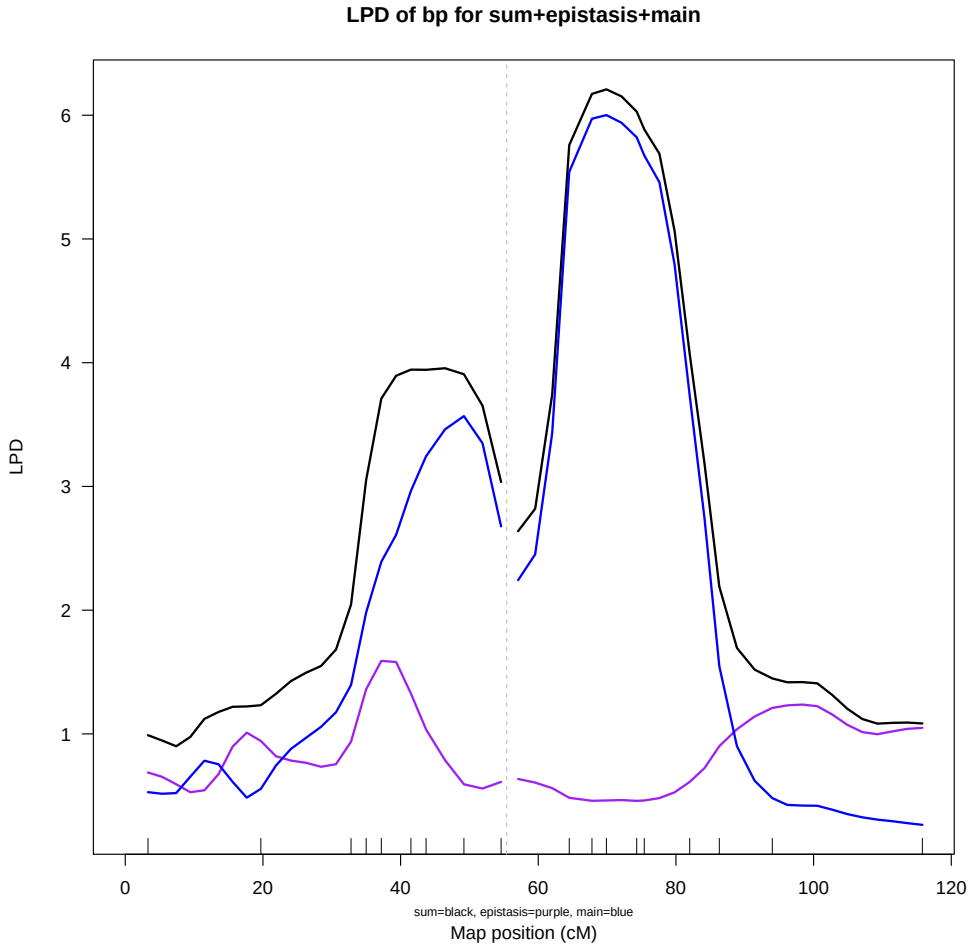
```
> qb.best(qbHyper)
```

```
Best pattern(s) by sq.atten score
    n.qtl chrom locus locus.LCL locus.UCL variance variance.LCL variance.UCL
247 0.625   1.2  72.1    62.025     98.36     4.86       0.0701        10.20
245 0.630   4.1  29.5    12.171     37.00    10.50       0.1610        17.80
248 0.662     6  59.0    13.833     66.70     4.72       0.1410        10.40
246 0.786    15  19.5    13.100     55.70     2.67       0.0894         7.27

Summary by better patterns
                     terms   percent    score cluster
1.2,4.1,6,15,6:15        4 3.1666667 4.000000       1
1.2,4.1,5,6,15,6:15      5 0.4666667 4.000000       1
1.2,4.1,6,15,15,6:15     5 0.2666667 3.852144       1
1.2,2,4.1,6,15,6:15      5 0.7000000 3.838877       1
1.2,4.1,6,6,15,6:15      5 0.4333333 3.822012       1
1.1,1.2,4.1,6,15,6:15    5 0.5333333 3.799457       1
1.2,4.1                  2 1.3666667 2.000000       2
1.2,2,4.1                3 0.7000000 2.000000       2
1.2,4.1,19               3 0.2666667 2.000000       2
1.1,1.2,4.1              3 0.3333333 1.876341       3

Maximum number of QTL in architecture: 10
```

```
> one <- qb.scanone(qbHyper, type = "LPD")
> summary(one)
```

```
LPD of bp for main,epistasis,sum

     n.qtl   pos m.pos e.pos   main epistasis    sum
1.1 0.3077 43.70 43.70 41.50  3.244     1.327  3.943
1.2 1.0233 67.80 67.80 67.80  5.972     0.459  6.172
2   0.3477 51.90 51.90 42.63  2.011     0.492  2.396
3   0.1453 30.63 30.63  8.76  1.145     3.068  1.678
4.1 1.1040 29.50 29.50 29.50 11.329     0.377 11.453
4.2 0.2730 74.30 74.30 74.30  0.717     4.884  5.336
5   0.2447 68.87 68.87 82.00  2.029     1.095  2.525
6   0.8383 59.00 59.00 59.00  3.745     5.959  9.069
7   0.1553 15.28 55.60 15.28  0.418     3.029  3.042
8   0.1320 56.93 59.00 17.52  0.946     1.626  1.488
9   0.1173 12.00 64.87 12.00  0.662     2.561  2.548
10  0.0947 37.95 75.40 37.95  0.581     0.840  0.984
11  0.1717 17.50 39.57 13.10  0.916     1.831  1.644
12  0.0947  1.10 46.55  1.10  0.452     2.197  2.368
13  0.0767 24.40 28.40 14.23  0.648     1.346  1.432
14  0.0840  0.00 46.35  0.00  0.621     2.059  2.310
15  0.9607 17.50 17.50 17.50  1.309     6.019  6.977
16  0.0813  8.37  8.37 10.46  0.396     1.710  1.744
17  0.1123 50.30  1.10 50.30  0.383     1.943  2.090
18  0.0663  2.20 14.20  2.20  0.599     2.070  2.245
19  0.1117 55.70 53.62 55.70  1.211     0.985  1.869
```

```
> plot(one, chr = 1)
```



4 Useful Plots and Summaries

A number of diagnostic routines are provided to assist with analysis. Some of these are bundled together in the generic `plot` routine for `qb` objects. For instance, `qb.scanone` and `qb.scantwo` can be used to identify the strength of main and epistatic QTL. All these routines have some connection to R/qtl (www.rqtl.org) routines, such as `scanone`, `scantwo` and `fitqtl`.

4.1 Plotting MCMC History

The `R/qtlbim` samples come from a Monte Carlo simulation. Are the MCMC samples well mixed? We can visually inspecting the history of the MCMC run. The command

```
> plot(qb.coda(qbHyper))
```

shows the MCMC chain as a time series. Each step, or iteration, of the MCMC chain represents a single model; therefore, we can explore the history of the MCMC chain by plotting time series for relevant model features. The time series plotted by `qb.coda` show the sampling histories for

1. number of QTL in each model model (`nqtl`),
2. mean phenotype according to each model (`mean`),
3. environmental variability under each model (`envvar`),
4. variance explained under each model (`var`) and

It is possible to plot a different subset of the model characteristics above, by using the optional argument `variables` in the `qb.coda` function. For example, in order to view just the number of QTL

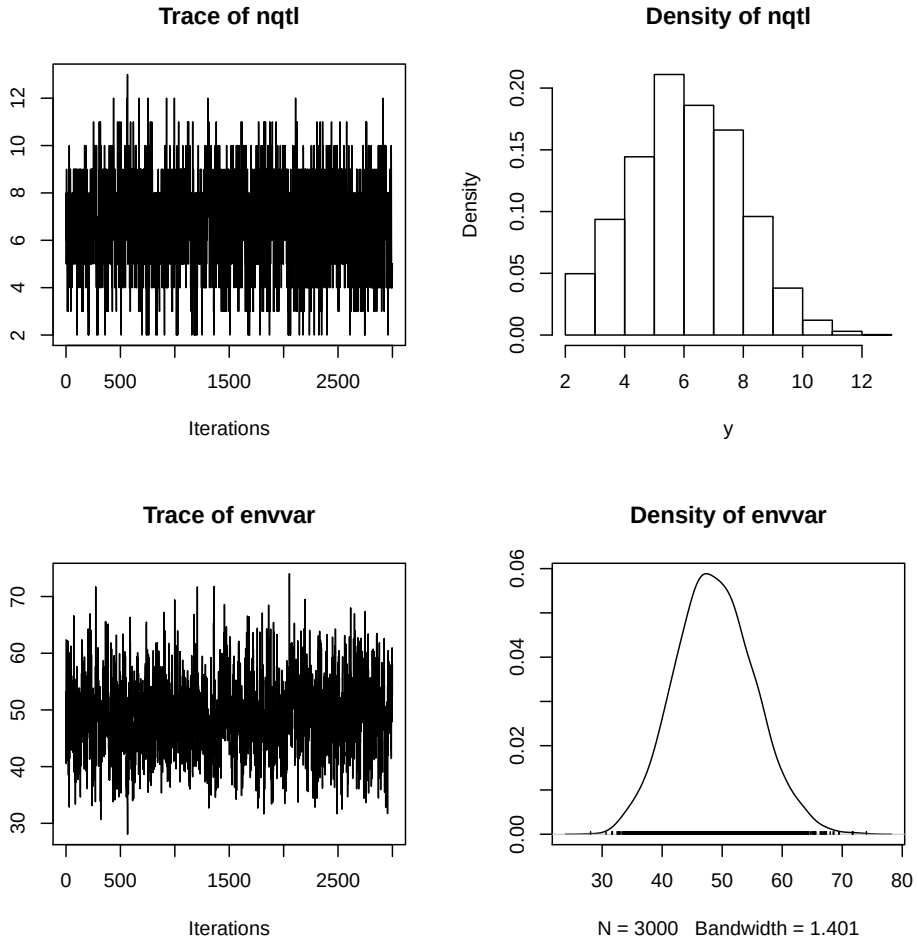


Figure 4: Diagnostic Plot for a MCMC run.

(`nqtl`) and the model means, use the following command. The results of the following command are shown in Figure 4.

```
> plot(qb.coda(qbHyper, variables = c("nqtl","envvar")))
```

4.2 A plot of sampled loci by chromosome

From a biological perspective it may be interesting to view the location of possible QTL along the chromosome. The function `qb.loci` shows a plot of quantitative trait loci for each chromosome. The QTL are from single QTL models appearing as samples in the MCMC chain. In the plot, the actual locations of possible QTL are jittered slightly in order to give a sense of the density of putative QTL in the vicinity of each marker. The code

```
> plot(qb.loci(qbHyper))
```

will produce a plot with all chromosomes. In order to view a subset of the chromosomes, the parameter `chr` to the generic `subset` routine can be used to limit the plot to a selected set of chromosomes. The horizontal (blue) lines in the plot show the locations of markers. The markers themselves can be labelled by using the parameters `markers` in the function.

```
> plot(qb.loci(subset(qbHyper, chr=c(3,4))), labels=TRUE)
```

Figure 5 shows the result of this command.

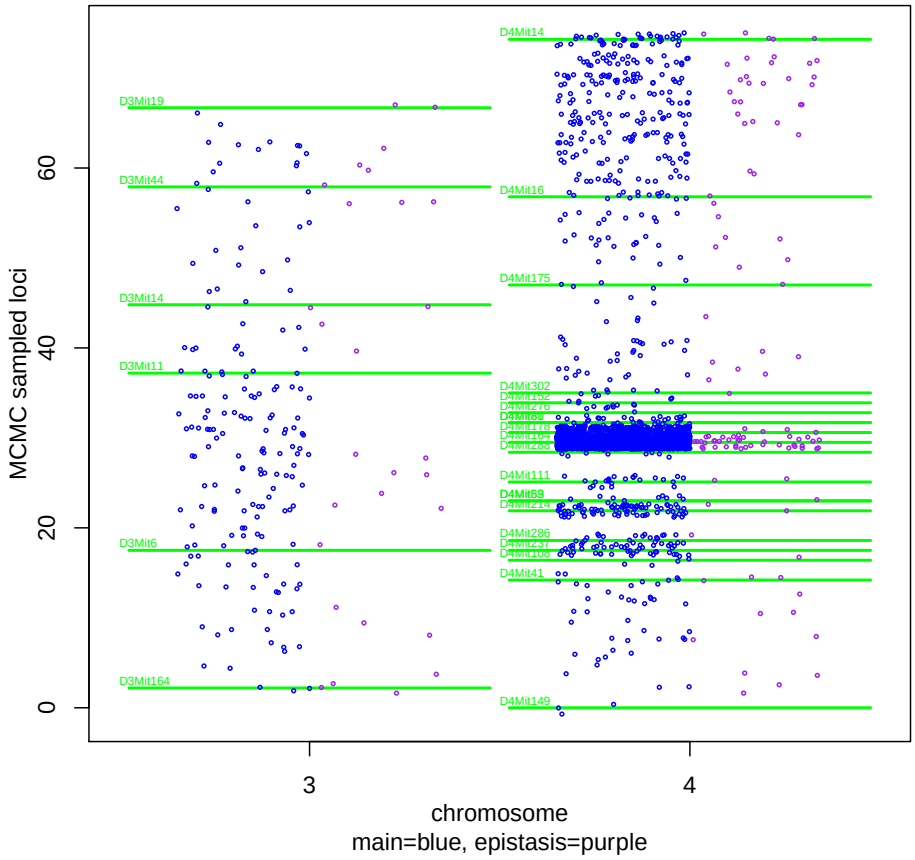


Figure 5: A jittered plot of quantitative trait loci, showing only only chromosomes 3 and 4, with locations and marker labels.

4.3 Bayes factor ratios

The function `qb.BayesFactor` produces a composite (4-by-2) summary plot of the models sampled by the MCMC chain. These plots are useful as an initial tool for examining the evidence in favor of multiple QTL models and in determining the locations of QTL. Figure 6 shows the plot produced by the command `qb.BayesFactor(qbHyper)`. The function of each of these plots is described below.

1. The plot appearing in the upper-left of the figure represents a plot of the prior distribution for the number of QTL involved in models (shown as a broken blue line) against the corresponding posterior probabilities (shown as a histogram).
2. The plot in the upper-right shows Bayes factor ratios. These are the ratios of posterior probabilities to prior probabilities. For pairs of values along the horizontal axis of this plot, the member of the pair with a larger Bayes factor ratio should be interpreted as more likely. The vertical arrows give an indication of the strength of evidence: weak (BF = 3), moderate (BF = 10) or strong (BF = 30).
3. The second row conveys information in terms of the pattern of chromosomes involved in the models.
4. The third row adresses the frequency of sampling each chromosome.
5. The fourth row show relative importance of epistatic pairs. Here the "6.15", or chr 6 by chr 15, epistatic pair is by far the strongest.

As with other plot functions in the `R/qtlbim` package, it is possible to limit attention of a subset of chromosomes using the generic `subset` routine. The `subset` argument `pattern` can be used to limit the models plotted to those involving a specified list of chromosomes. For example the command `qb.BayesFactor(subset(qbHyper,pattern=c(2,3,17)))` considers only those models involving chromosomes 2,3 and 17. Repeats in the pattern sequence indicate multiple QTL on the same chromosome.

4.4 Other plots of interest

An experimental plot uses highest posterior density (HPD) regions. The profile of the posterior is interpreted as a density, and the smallest region containing 50% (by default) of the density is the HPD region. The command `plot(qb.hpdone(qbHyper))` yields Figure 7.

Coefficients for epistatic effects for the most probable epistatic pairs are shown in Figure 8. `plot(qb.epistasis(qbHyper))`. produces jittered plots of sampled Cockerham effects, overlaid with boxplots. Summaries are provided as well but not shown here.

Summary diagnostics as histograms and boxplots by number of QTL. This diagnostic plot can be generated by the command `plot(qb.diag(qbHyper))`, as shown in Figure 9.

5 Data Management

5.1 Data Simulation

R/qtlbim has an inbuilt function `qb.sim.cross` to simulated a backcross or F2 data set of class `cross` (see R/qtl help pages for details). The following chunk of code generates a data set of 100 individuals of F2 mating design. These individuals are genotyped for 11 not equally spaced markers on 20 chromosomes. There are 7 QTLs, two on chromosome 1 and one each on chromosomes 3,5,7,10 and 19. QTL numbers 1,3 and 4 have additive main effects of 0.5, -0.5 and 0.5 and numbers 2 and 4 have dominant main effects of 0.5 and -0.5. QTL numbers 4 and 5 have an additive-additive interaction of -0.7 and numbers 6 and 7 have an additive-dominant interaction of 1.2. Two covariates, a binary fixed covariate and an ordinal random are generated with their corresponding coefficients as 0.5 and 0.07. G x E (gene x environemt) interaction is also considered with the fixed covariate. A normal phenotype and an ordinal phenotype with 3 categories are measured. 7% of the genotypes are randomly missing.

```
> cross <- qb.sim.cross(len = rep(100, 20), n.mar = 11, eq.spacing = F,
+     n.ind = 100, type = "f2", ordinal = c(0.3, 0.3, 0.2, 0.2),
+     missing.geno = 0.03, missing.pheno = 0.07, qtl.pos = rbind(c(1,
+         15), c(1, 45), c(3, 12), c(5, 15), c(7, 15), c(10, 15),
+         c(12, 35), c(19, 15)), qtl.main = rbind(c(1, 0.5, 0),
+         c(2, 0, 0.7), c(3, -0.5, 0), c(4, 0.5, -0.5)), qtl.epis = rbind(c(4,
+         5, -0.7, 0, 0, 0), c(6, 8, 0, 1.2, 0, 0)), covariate = c(0.5,
+         0.07), gbye = rbind(c(7, 0.8, 0)))
```

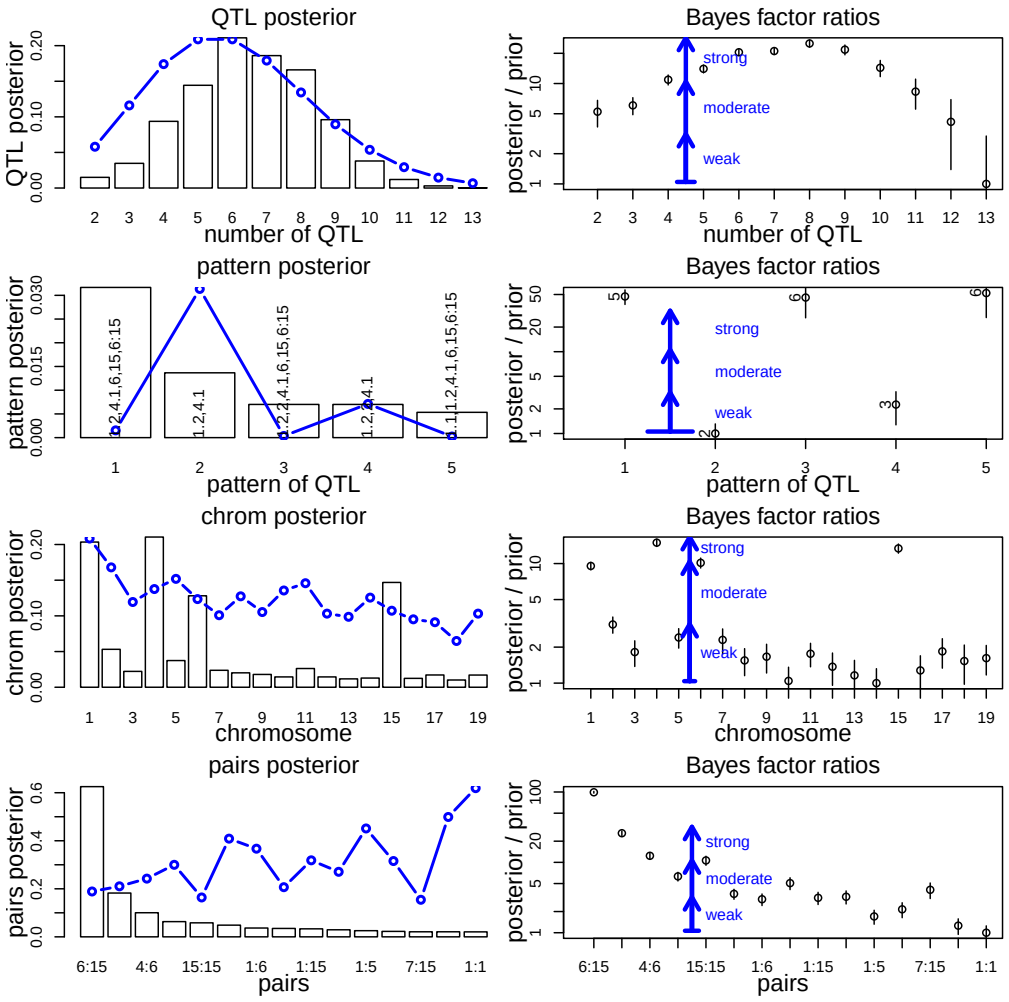


Figure 6: Paired plots of posteriors as bars overlaid by priors as blue lines (left panels) with Bayes factor ratios to the least likely model (right panels). Models in right panel can be compared by vertical separation as scale is geometric. Blue arrows on right panels indicate weak, moderate or strong Bayes factors for ratios of 3, 10 or 30, respectively. Rows convey information about (1) number of QTL, (2) chromosome pattern of QTL, (3) chromosomes, (4) epistatic pairs.
strong

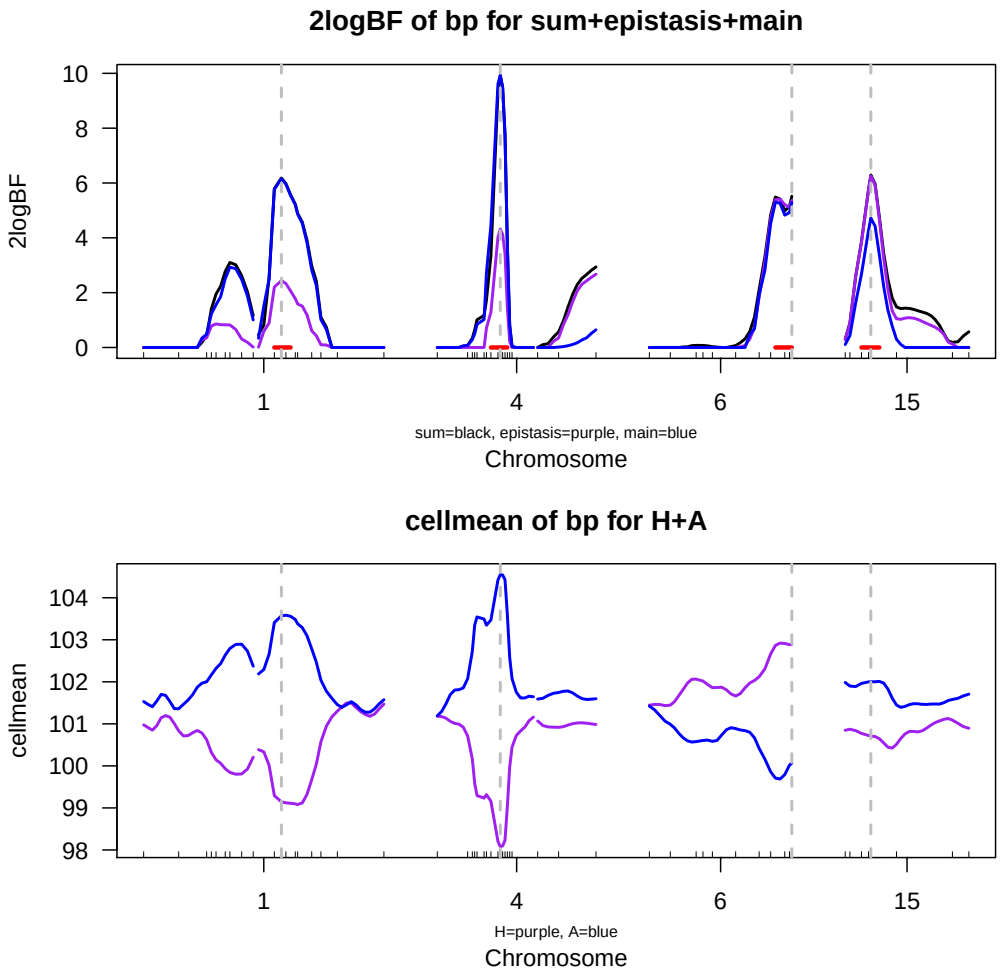


Figure 7: A paired plot of posterior scan for loci above a scan of marginal genotypic means by locus. In upper panel, black is overall posterior, blue is for main effects and purple is for epistasis. In lower panel, blue is for AA, purple for AB, and red for BB genotype at scanned locus.

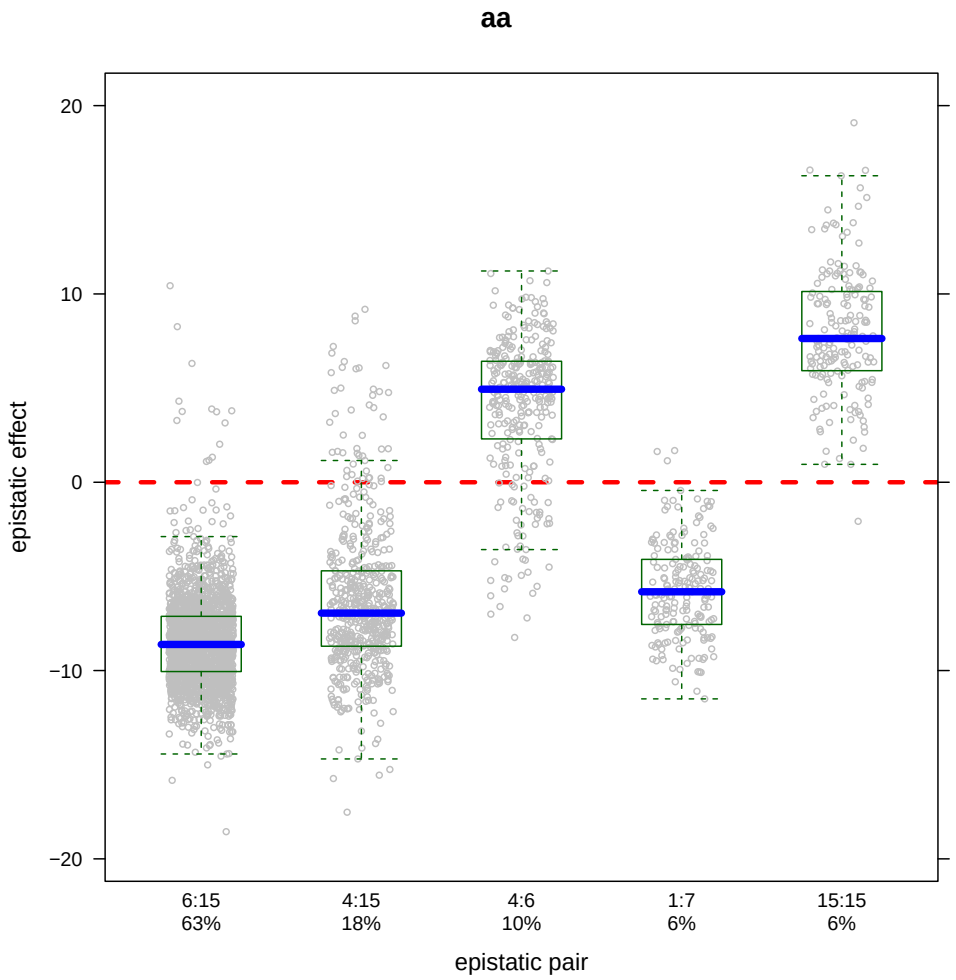


Figure 8: A plot of epistatic effect by pair using Cockerham effects. Only stronger epistatic pairs are shown. Blue line at median; box contains 50% of samples for epistatic pair. Percent below pair indicates percent of MCMC samples with this epistatic pair.

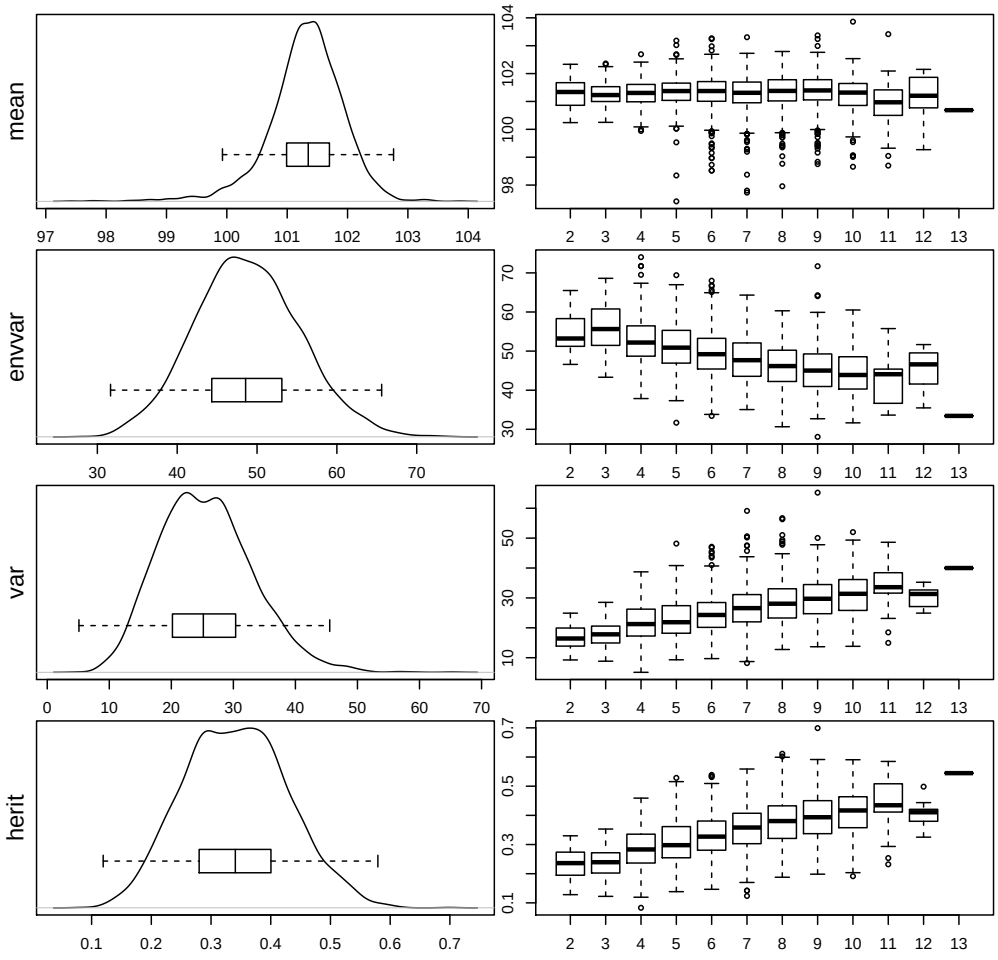


Figure 9: A set of diagnostic plots. Default has mean, unexplained variance (`"envvar"`), explained variance (`"var"`), and heritability (`"herit"`). Left panels show density plot and horizontal box plot for all samples. Right panels show box plots by number of QTL.

By using the function `qb.sim.cross` a list is attached to cross object named "qtl". This list is typically not a part of the `cross` object as described in `read.cross` of the R/qtl library and is generated only with the `qb.sim.cross()` function.

```
> names(cross)
```

```
[1] "geno"   "pheno"  "qtl"    "gvalue"
```

The `cross$gvalue` is a vector of predicted values of the same length as the phenotype `cross$pheno$pheno.normal`. The `cross$qtl` contains information about the true values which can be compared to after the analysis.

```
> summary(cross$qtl)
```

```
$pos
      chr pos
qtl.1   1  15
qtl.2   1  45
qtl.3   3  12
qtl.4   5  15
qtl.5   7  15
qtl.6  10  15
qtl.7  12  35
qtl.8  19  15

$herit.main
       qtl        add        dom
main.1   1 0.05608653 0.00000000
main.2   2 0.00000000 0.05496480
main.3   3 0.05608653 0.00000000
main.4   4 0.05608653 0.02804326

$herit.epis
       qtl.a qtl.b        aa        ad da dd
epis.1     4     5 0.0549648 0.0000000  0  0
epis.2     6     8 0.0000000 0.0807646  0  0

$herit.cov
        fix.cov    ran.cov
[1,] 0.02804326 0.03140846

$herit.gbye
      qtl        add dom
GxE.1   7 0.03589538   0
```

The summary of the cross object summary is shown below.

```
> summary(cross)
```

```
    F2 intercross

    No. individuals:    100

    No. phenotypes:     4
    Percent phenotyped: 93 94 95 91

    No. chromosomes:    20
        Autosomes:      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

    Total markers:      220
    No. markers:        11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
                        11
    Percent genotyped:  96.9
    Genotypes (%):      AA:24.2  AB:51.1  BB:24.8  not BB:0  not AA:0
```

6 Theoretical Development

This section could be skipped. It is aimed at those quantitative folks who have read Yi et al. (2005) for the math and want to know more. Here we leave out details concerning covariates to simplify presentation.

Given complete data on genotypes for all individuals across the genome, we could consider a model relating phenotype y to genotype g through a design matrix X,

$$y = \mu + X\Gamma\beta + e \ .$$

The unknown effect parameters are the grand mean, μ, the effect parameters, β, and the unexplained variance, $\sigma^2 = V(e)$, which for convenience, we bring together as $\theta = (\mu, \beta, \sigma^2)$. The genetic architecture is specified by $\Gamma = \text{diag}(\gamma)$, which has values of 1 or 0 to indicate presence or absence, respectively, of the corresponding model effect. The QTL model could thus be written as $p(y|\gamma, X, \theta)$. [For practical purposes, the maximum number of QTL is rarely over $l_0 + 3\sqrt{l_0}$ where l_0 is the prior mean for the total number of QTL. Hence, the size of X stored at any time can be greatly reduced through careful bookkeeping. See Yi et al. (2005) for details.]

Here are some details on the Cockerham epistatic model for experimental crosses with *K+1* genotypes per loci ($K = 1$ for backcross, $K = 2$ for intercross). There are K main effects and K^2 epistatic effects. For a backcross population with two segregating genotypes, bb and Bb, at locus, the main effect has predictor $x_1 = z - 0.5$, where z denotes the number of b alleles. The epistatic effect predictors are products of main effect predictors. An intercross has three segregating genotypes bb, Bb and BB at any locus. The two main effect predictors for additive and dominance in the Cockerham model are $x_1 = z - 1$ and $x_2 = (1 - \text{abs}(x_1)) - 0.5$, respectively. The four epistatic effect predictors for a pair of loci are cross products of the main effect predictors.

This genetic architecture, specified by a 0-1 vector γ, allows us to consider models of different dimensions, e.g. one vs. two QTL, without resorting to a more complicated (reversible jump) sampling scheme. The unknown values γ are the key device in sampling over many different possible genetic architectures, in terms of what loci λ are included and what gene action is important. There is some redundancy between γ and λ: a locus is in the model only if at least one γ associated with that locus is 1. Technically, we consider probabilities $p(\lambda|\gamma)$ that can only be 0 or 1 to indicate whether the loci, λ, are compatible with the genetic architecture, γ. While the loci are determined by the genetic architecture, γ is not completely determined by λ. We exploit this to make more efficient code and to build diagnostic summaries.

6.1 Likelihood and posterior

In a classical setting, the full likelihood augmented by genotypes, g, over the genome is

$$p(y, g|m, \gamma, \theta) = p(y|\gamma, X, \theta)p(X|g)p(g|m, \lambda)p(\lambda|\gamma) \ ,$$

with m the marker genotypes across the genome and $p(g|m, \lambda)$ the map function. The whole-genome genotype information, g, and the design matrix, X, are 1-1 mappings: $p(X|g)$ is either 1 or 0, depending on whether or not the design is compatible with the genotypes. At most loci, we do not fully know genotypes g, hence the likelihood given observable data is averaged over g,

$$L(\gamma, \theta|y, m) = \sum_g p(y, g|m, \gamma, \theta) \ .$$

With no QTL, we write $L(\mu|y)$ for the null likelihood.

In a Bayesian perspective, a prior $p(\gamma, \theta)$ is placed on the unknowns, and we study the posterior,

$$p(g, \gamma, \theta|y, m) \propto p(y, g|m, \gamma, \theta)p(\gamma, \theta) \ .$$

To study the unknown parameters of interest, (γ, θ), we average the posterior over the genotypes, or equivalently, form a weighted average of the augmented likelihood with weights proportional to the prior on (γ, θ),

$$p(\gamma, \theta|y, m) = \sum_g p(g, \gamma, \theta|y, m) \propto \sum_g p(y, g|m, \gamma, \theta)p(\gamma, \theta) \ .$$

6.2 Parameter estimation

Classically, the parameters of interest, (λ, θ), are estimated by maximizing the likelihood. This is usually done in a QTL setting by profiling the likelihood, or LOD (see below), with respect to one locus or two loci over the genome. We think of that here as profiling with respect to a given genetic architecture, γ, to find the maximum likelihood estimate (MLE) for β ,

$$\hat{\beta} = V\Gamma X^T y\,,$$

with $V = (\Gamma X^T X \Gamma)^{-1}$ and $\sigma^2 V$ the variance-covariance matrix for $\hat{\beta}$. Here we assume the columns of X are centered on zero, so the MLE for the reference is $\hat{\mu} = \bar{y}$.

Bayesian parameter estimates are typically found as the posterior means, which shrink $\hat{\mu}$ toward its prior mean μ_0 and $\hat{\beta}$ toward the prior mean of 0, leading to posteriors

$$\mu \sim N\left((1-b)\mu_0 + b\bar{y}, b\sigma^2/n.ind\right)\ ,$$

and

$$\beta \sim N\left(B\hat{\beta}, B\sigma^2 V\right)\ ,$$

with b and B being Bayesian shrinkage factors. As we gather more data, the Bayesian priors focus on the MLEs, i.e. b and B tend to 1. The likelihood and the posterior are both fairly symmetric around the maximum, for any given γ. Thus, the posterior mean and the MLE for β are very close in practice. This is less apparent from the summaries in the previous section, as the Bayesian estimates are attenuated by the putative effects of other QTL along the genome. This is a technical post-processing issue of properly sorting out the effects of multiple linked loci.

6.3 Variance components

Variance components can also be estimated in both approaches. The classical unbiased estimate for environmental variance is $\hat{\sigma}^2 = RSS(\hat{\theta})/df$, with $RSS(\theta) = \sum(y - \mu - X\Gamma\beta)^2$ and $df = n.ind - 1 - \sum\gamma$.

A Bayesian posterior estimate of σ^2 is its posterior mean, which is a weighted average of $RSS(\theta)/n.ind$ and its prior mean. Its empirical estimate can be found by averaging the posterior samples,

```
> summary(qb.scanone(qbHyper, type = "variance", scan = "env"))

variance of bp for env

     n.qtl   pos m.pos  env
1.1 0.3077 43.70 43.70 48.7
1.2 1.0233 67.80 67.80 49.1
2   0.3477 51.90 51.90 49.3
3   0.1453 30.63 30.63 48.2
4.1 1.1040 29.50 29.50 48.7
4.2 0.2730 74.30 74.30 44.0
5   0.2447 68.87 68.87 47.2
6   0.8383 59.00 59.00 46.0
7   0.1553 15.28 55.60 48.2
8   0.1320 56.93 59.00 49.6
9   0.1173 12.00 64.87 48.4
10  0.0947 37.95 75.40 49.7
11  0.1717 17.50 39.57 47.6
12  0.0947  1.10 46.55 47.0
13  0.0767 24.40 28.40 48.0
14  0.0840  0.00 46.35 48.2
15  0.9607 17.50 17.50 46.4
16  0.0813  8.37  8.37 49.9
17  0.1123 50.30  1.10 46.9
18  0.0663  2.20 14.20 48.0
19  0.1117 55.70 53.62 47.3
```

Heritability is computed as the percent of explained variation, $h^2 = 100(TSS - RSS(\theta))/TSS$, with $TSS = \sum(y - \bar{y})^2$ the total sum of squares. [The idealized variation would substitute expected fractions for the X^2 terms based on the type of cross.] We can find the posterior estimate of variability as the `main` entry below:

```
> summary(qb.scanone(qbHyper, type = "heritability"))

heritability of bp for main,epistasis,sum

     n.qtl   pos m.pos e.pos   main epistasis   sum
1.1 0.3077 43.70 43.70 41.50  5.210     1.867  6.45
1.2 1.0233 67.80 67.80 67.80  9.777     0.385 10.14
2   0.3477 51.90 51.90 42.63  3.133     0.464  3.82
3   0.1453 30.63 30.63  8.76  1.614     4.946  2.56
4.1 1.1040 29.50 29.50 29.50 18.055     0.226 18.28
4.2 0.2730 74.30 74.30 74.30  0.805     7.924  8.73
5   0.2447 68.87 68.87 82.00  3.151     1.521  4.03
6   0.8383 59.00 59.00 59.00  5.525     9.188 14.62
7   0.1553 15.28 55.60 15.28  0.336     4.899  4.92
8   0.1320 56.93 59.00 17.52  1.272     2.475  2.24
9   0.1173 12.00 64.87 12.00  0.771     4.097  4.08
10  0.0947 37.95 75.40 37.95  0.627     1.092  1.35
11  0.1717 17.50 39.57 13.10  1.217     2.822  2.50
12  0.0947  1.10 46.55  1.10  0.400     3.457  3.76
13  0.0767 24.40 28.40 14.23  0.746     1.990  2.14
14  0.0840  0.00 46.35  0.00  0.701     3.200  3.65
15  0.9607 17.50 17.50 17.50  1.736     9.677 11.38
16  0.0813  8.37  8.37 10.46  0.302     2.618  2.68
17  0.1123 50.30  1.10 50.30  0.288     3.019  3.28
18  0.0663  2.20 14.20  2.20  0.662     3.235  3.55
19  0.1117 55.70 53.62 55.70  1.723     1.329  2.89
```

6.4 LOD, LPD and BF

The classical approach introduced by Lander and Botstein (1989) profiles the likelihood only along the ridge of maximum β for each λ. That is, at each λ, find β that maximizes the LOD. The LOD map is a plot of this profile. The LOD statistic to assess QTL is

$$LOD(\lambda) = c + \log_{10}\left(\max_{\theta} L(\gamma, \theta|y, m)p(\lambda|\gamma)\right) ,$$

with the constant being $c = -\log_{10}(\max_{\mu} L(\mu|y))$. The likelihood ratio is $LR = 10^{LOD}$, and deviance is $D = 2\log(10)LOD$.

The Bayesian approach provides a direct estimate of the posterior as the histogram of the samples from the Markov chain Monte Carlo. Sen and Churchill (2002) proposed profiling the log posterior density, LPD, which involves averaging over the unknown parameters θ,

$$LPD(\lambda) = C + \log_{10}\left(\sum_{\theta} p(\gamma, \theta|y, m)p(\lambda|\gamma)\right) .$$

[The sum over θ is actually an multidimensional integral, but we ignore those details here.] Here the constant C would involve averaging over the null likelihood with respect to the prior on μ. In practice, LOD and LPD are often pretty close to each other and can be used interchangeably.

One advantage of sampling a large set of possible models by MCMC is that Bayes factors are easily computed. We do not have to resort to fancy harmonic means as in Newton and Raftery (199x). Instead, we construct marginal posterior histograms for models to be compared, and rescale by their priors. For instance, to compare two genetic architectures, we construct

$$BF = \frac{p(\gamma|y, m)/p(\gamma)}{p(0|y)/p(0)} ,$$

in which $p(0)$ is the prior on γ being all zero (no QTL at all) and $p(0|y)$ is the posterior. Actually, $p(0|y)/p(0) \propto p(y) = \sum_{\mu} p(y|\mu)p(\mu)$, with the sum really an integral over the real line. Often this is more interpretable on a log scale as $2\log(BF)$, which we can compute as

```
> summary(qb.scanone(qbHyper, type = "2logBF"))
```

```
2logBF of bp for main,epistasis,sum

     n.qtl   pos m.pos e.pos  main epistasis    sum
1.1 0.3077 43.70 43.70 41.50 2.927     0.827 3.1015
1.2 1.0233 67.80 67.80 67.80 6.157     2.438 6.1807
2   0.3477 51.90 51.90 42.63 1.117     0.000 1.3074
3   0.1453 30.63 30.63  8.76 0.000     0.000 0.0000
4.1 1.1040 29.50 29.50 29.50 9.923     4.317 9.9240
4.2 0.2730 74.30 74.30 74.30 0.644     2.673 2.9394
5   0.2447 68.87 68.87 82.00 1.357     0.370 1.5288
6   0.8383 59.00 59.00 59.00 5.296     5.389 5.4882
7   0.1553 15.28 55.60 15.28 0.000     0.293 0.3432
8   0.1320 56.93 59.00 17.52 0.000     0.000 0.0000
9   0.1173 12.00 64.87 12.00 0.000     0.000 0.0000
10  0.0947 37.95 75.40 37.95 0.000     0.000 0.0000
11  0.1717 17.50 39.57 13.10 0.000     0.000 0.0643
12  0.0947  1.10 46.55  1.10 0.000     0.000 0.0000
13  0.0767 24.40 28.40 14.23 0.000     0.000 0.0000
14  0.0840  0.00 46.35  0.00 0.000     0.000 0.0000
15  0.9607 17.50 17.50 17.50 4.718     6.257 6.2913
16  0.0813  8.37  8.37 10.46 0.000     0.000 0.0000
17  0.1123 50.30  1.10 50.30 0.000     0.000 0.0000
18  0.0663  2.20 14.20  2.20 0.000     0.000 0.0000
19  0.1117 55.70 53.62 55.70 0.000     0.000 0.0000
```

6.5 Marginal Summaries

Our primary interest here is in marginal statistics. Consider that the model has genetic architecture γ that include loci λ. We want to ask what is the contribution to the model of some subset of indicators, γ_2, associated with a locus, or a set of loci, λ_2. We might ask this in a variety of ways, looking at evidence in terms of LOD or a related statistics, or the contribution in terms of variance components, heritability, or parameter effects. We can think of partitioning the genetic architecture into two components, $\gamma = (\gamma_1, \gamma_2)$, with a corresponding partition of the effect parameters,

$$\Gamma\beta = (\Gamma_1 + \Gamma_2)\beta \ .$$

The subset of effect parameters, $\beta_2 = \Gamma_2\beta$, may include, for instance, the main effects for locus λ_2 plus some or all epistatic effects that involve this locus. We can then ask questions about β_2, or about γ_2 and λ_2, adjusting for the presence of effects $\beta_1 = \Gamma_1\beta$. Note that β_1 could include some model parameters for λ_2.

6.5.1 Variance components

Here and through the rest of this document, we argue that we can characterize important diagnostic summaries using marginal properties of MCMC samples. The key technical argument is in the next paragraph. Namely, we can use the marginal variance components of our model fit, ignoring covariances, to construct approximate statistics.

If the columns of X are nearly orthogonal to each other, then the variance-covariance matrix for the effect parameter MLEs, $\text{var}(\hat{\beta}) = \sigma^2 V$, would be *diagonally dominant*. That is, we suppose the variances along the diagonal are larger than the sum of the absolute covariances. Formally, with $v = \text{diag}(V)$ and $V_{(j)}$ the j column of V,

$$2v_{(j)} \geq \sum |V_{(j)}| \ .$$

In other words, we assume the covariances among effect estimates are negligible, and the diagonal values are approximately $v_{(j)} \approx \gamma_{(j)} / \sum X_{(j)}^2$, with $X_{(j)}$ the jth column of X. In this case we can approximate V by its diagonal, $D = \text{diag}(v)$, and get a good approximation of V^{-1} using D^{-1}:

$$V^{-1} = D^{-1}[I + O]^{-1} \ ,$$

with O being on the order of $(V - D)D^{-1}$. As long as the diagonal entries of D are large, then this approximation is good. Where these variances are small, the approximation is not so useful.

Since we are interested in learning about effects with larger variance components, this approximation seems quite workable in the present setting. It should a pretty reasonable between terms for unlinked loci, and under conditions of Hardy-Weinberg equilibrium among alleles at each locus. Note also that epistatic effects between linked loci will be addressed directly by construction of columns of X. [I believe the discrepancy of the diagonal can be readily checked under H-W by adding another `type` to the `qb.scan` routines–next freeze.]

With this approximation the explained variation can be approximated as

$$TSS - RSS(\theta) = \sum (X\Gamma\beta)^2 \approx \gamma^T r \,,$$

with $r_{(j)} = \beta^2_{(j)} \sum X^2_{(j)}$ being the variance explained by the jth component of the genetic architecture. Then the difference, $RSS(\theta_1) - RSS(\theta) \approx \gamma_2^T r = \sum r_2$, is simply the sum of variance components, which are readily stored for each MCMC iteration. Here, r_2 contains the elements of r corresponding to $\gamma_2 = 1$, and $\theta_1 = (\mu, \beta_1, \sigma^2)$.

Marginal heritability is computed as the additional variation explained by the genetic architecture γ_2 given γ_1 ,

$$h^2 = \frac{RSS(\theta_1) - RSS(\theta)}{TSS} = \frac{\gamma_2^T r}{TSS} \,.$$

6.5.2 LOD, LPD and BF

The adjusted LOD to compare the full model to the reduced model with $\gamma_2 = 0$ is

$$LOD(\gamma_2|\gamma_1) = \log_{10}\left(\frac{\max\limits_{\theta} L(\gamma, \theta|y, m)}{\max\limits_{\theta_1} L(\gamma_1, \theta_1|y, m)}\right) \,.$$

The adjusted LPD is similarly,

$$LPD(\gamma_2|\gamma_1) = \log_{10}\left(\sum_{\theta} \frac{p(\gamma, \theta|y, m)}{p(\gamma_1, \theta_1|y, m)}\right) \,,$$

with again the sum actually being an integral over θ .

In the case of normal data and complete marker information, the LOD reduces to

$$LOD(\gamma_2|\gamma_1) = \frac{n.ind}{2} \log_{10}\left(\frac{\min\limits_{\theta_1} RSS(\theta_1)/df_1}{\min\limits_{\theta} RSS(\theta)/df}\right) \,,$$

with degrees of freedom, $df = n.ind - 1 - \sum \gamma$, and $df_1 = n.ind - 1 - \sum \gamma_1$. The LPD follows a similar form, but involving an average (or really, integral) over θ ,

$$LPD(\gamma_2|\gamma_1) = \frac{n.ind}{2} \log_{10}\left(\sum_{\theta} \frac{RSS(\theta_1)/df_1}{RSS(\theta)/df}\right) \,.$$

The Bayes factors are easily computed, as noted earlier. To compare the two genetic architectures γ and γ_1 , we construct

$$BF = \frac{p(\gamma|y, m)/p(\gamma)}{p(\gamma_1|y, m)/p(\gamma_1)} \,.$$

Often this is more interpretable on a log scale as $2\log(BF)$, which we can compute as

6.6 Model Averaging Algorithm

Here we briefly describe the model averaging idea. The MCMC samples include a wide variety of models, indexed by γ. The 1-D and 2-D scans first compile a selected diagnostic for each sample (also known as an iteration). That is, at each genome position, or pair of positions, we average the values for samples that include that position, i.e. have $\gamma = 1$ at that position. The posterior is simply an average of the γ samples at each position.

These samples are kept for each model component, either in terms of the un-aggregated Cockerham (1954) partition or in terms of `main` effects and `epistasis`, and for the `sum` of these components. There

are some mechanics involved. For instance, for 1-D averages involving epistasis, we want to count each pair for both loci, and for 2-D averages, we want to count epistatic effects separately at each locus. But these are details that can be found by looking at the code if interested.

Chromosome summaries, or summaries within regions of chromosomes, are found as weighted averages of these per-position summaries. The weights are naturally the number of MCMC samples per position. At present the code does not separate out multiple loci on a chromosome [next freeze].

With moderate MCMC sample sizes, the 1-D and 2-D scans can be rather rough, or jagged. We have found nearest neighbor smoothing to be helpful. That is, a position is equally weighted against the sum of its neighbors, accounting for number of MCMC samples. This can be repeated several times (e.g. `smooth = 3`) to further local smoothing.

7 Summary

In this overview, we have explored the use of many of the Bayesian interval mapping routines. Through examples using the `hyper` experimental data, we have demonstrated the key steps in identifying both main and epistatic effects. Further information on using using `R/qtlbim` to explore the `hyper` data set can be found in the `prototype.qtl.hyper.slides` vignette. In order to view the vignette, simply type

```
> vignette(topic="prototype.qtl.hyper.slides",package="qtlbim")
```

at the R prompt.

References

Broman KW (2001) Review of statistical methods for QTL mapping in experimental crosses. Lab Animal 30(7): 44-52.

Broman KW, Wu H, Sen S, Churchill GA (2003) R/qtl: QTL mapping in experimental crosses. Bioinformatics 19: 889-890.

Cockerham CC (1954) An extension of the concept of partitioning hereditary variance for analysis of covariances among relatives when epistasis is present. Genetics 39: 859-882.

Haley C, Knott S (1992) A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. Heredity 69: 315-324.

Jiang C, Zeng ZB (1997) Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. Genetica 101: 47-58.

Kao CH, Zeng ZB (2002) Modeling epistasis of quantitative trait loci using Cockerham's model. Genetics 160: 1243-1261.

Manichaikul A, Moon JY, Sen S, Yandell BS, Broman KW (2008) A model selection approach for the identification of quantitative trait loci in experimental crosses. Genetics (submitted).

Morton NE (1995) LODs past and present. Genetics 140: 7-12.

Newton MA, Raftery AE (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with Discussion). Journal of the Royal Statistical Society, series B, 56, 3-48.

Sen S, Churchill GA (2001) A statistical framework for quantitative trait mapping. Genetics 159: 371-387.

Sugiyama F, Churchill GA, Higgens DC, Johns C, Makaritsis KP, Gavras H, Paigen B (2001) Concordance of murine quantitative trait loci for salt-induced hypertension with rat and human loci. Genomics 71: 70-77.

Wright FA, Kong A (1997) Linkage mapping in experimental crosses: the robustness of single-gene models. Genetics 146: 417-425.

Yandell BS, Mehta T, Banerjee S, Shriner D, Venkataraman R, Moon JY, Neely WW, Wu H, von Smith R, Yi N (2007) R/qtlbim: QTL with Bayesian interval mapping in experimental crosses. Bioinformatics 23: 641-643.

Yi N, Banerjee S, Shriner D, Pomp D, Yandell BS (2007a) Bayesian mapping of genome-wide interacting QTL for ordinal traits. Genetics 176: 1855-1864.

Yi N, Shriner D, Banerjee S, Mehta T, Pomp D, Yandell BS (2007b) Efficient strategies for Bayesian mapping of genome-wide interacting QTL. Genetics 176: 1865-1877.

Yi N, Yandell BS, Churchill GA, Allison DB, Eisen EJ, Pomp D (2005) Bayesian model selection for genome-wide epistatic quantitative trait loci analysis. Genetics 170: 1333-1344.