Model Selection of Genetic Architecture using **R/qtlbim**

Brian S. Yandell, Jee Young Moon

April 11, 2008

Abstract

R/qtlbim (www.qtlbim.org) provides a powerful suite of tools for model selection of genetic architecture. The Markov chain Monte Carlo (MCMC) sampling approach draws samples from the more probable genetic architectures. Subsequent visualization and summary of these MCMC samples can inform users about the most probable genetic architecture. The tools described herein were developed largely in 2007 to augment or extend tools already in R/qtlbim.

1 Overview

This vignette describes the model selection routines for MCMC samples already obtained. using previously described tools in the `R/qtlbim` package. The purpose of these plots and summaries is to help users select the best or better models to explain the relationship between phenotype and genotype. We focus on the `hyper` data set, and more particularly on the MCMC samples already generated, `qbHyper`.

```
> library(qtlbim)
> data(qbHyper)
```

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`);

This document assumes familiarity with the `hyper` analysis using R/qtlbim, as well as with the basics of this package. Please see the other vignettes for further package details.

2 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 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
```

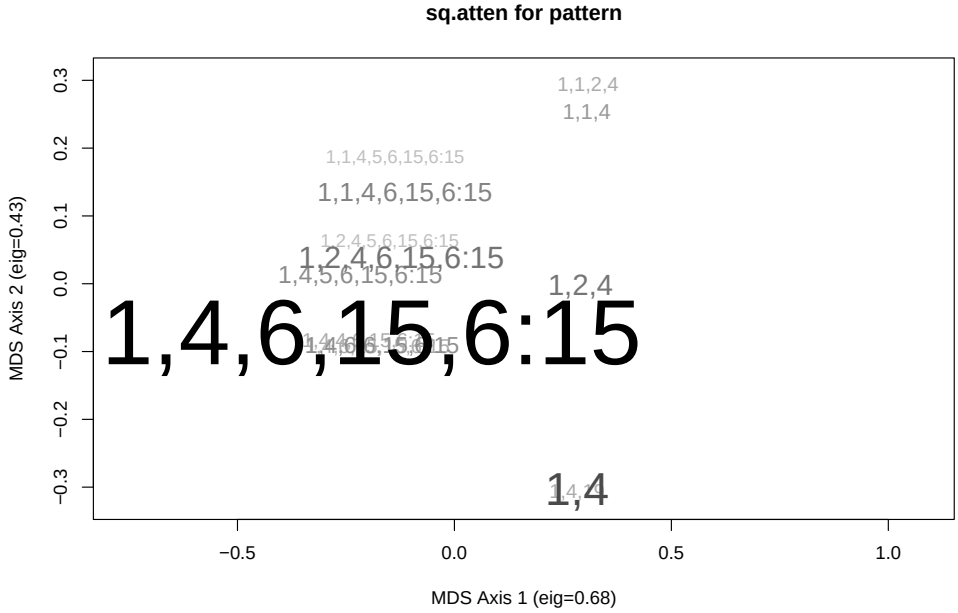
The pattern with the highest posterior probability is 1,4,6,15,6:15, whereas the pattern with highest Bayes factor is 1,4,4,6,15,6:15. Patterns are represented a chromosome identifiers separated by commas; epistatic pairs of chromosomes are joined by a colon. The `qb.bf` summary model-averages over all 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` can perform this task, and a few more.

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

Maximum number of QTL in architecture: 11

Summary by pattern
                  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

Best pattern(s) by sq.atten score
    chrom locus variance locus.LCL locus.UCL variance.LCL variance.UCL n.qtl
247     1  69.9 4.331837  24.44875   95.7985   0.03452814     9.871876  2408
245     4  29.5 9.098802  14.20000   74.3000   0.08845976    17.239369  2640
248     6  59.0 4.725800  13.83333   66.7000   0.12963260    10.517350  2129
246    15  19.5 2.638343  13.10000   55.7000   0.08227567     7.310082  2535
```

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)
```

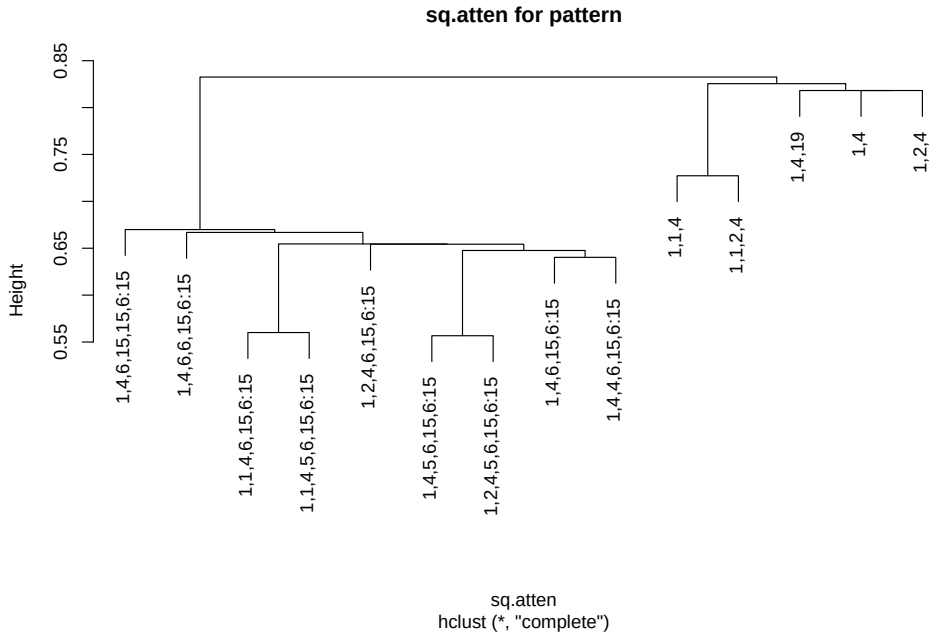


The font size of a pattern is determined by its posterior probability. The 2-D multidimensional scaling (MDS) projection is based on the `score.type` (see below). Notice that models that overlap with 1,4,6,15,6:15 are plotted near that pattern. Other patterns with little overlay are some distance away.

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]]
```

	chrom	locus	variance	locus.LCL	locus.UCL	variance.LCL	variance.UCL	n.qtl
247	1	69.9	4.291848	24.06667	96.18000	0.03516970	10.027673	3993
245	4	29.5	9.206616	14.20000	74.30000	0.08047250	17.222186	4131
248	6	59.0	4.065665	9.80000	66.70000	0.04463393	10.274912	2515
246	15	19.5	2.442734	13.10000	58.26667	0.04279294	7.205367	2882

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

	chrom	locus	variance	locus.LCL	locus.UCL	variance.LCL	variance.UCL	n.qtl
247	1	69.9	4.331837	24.44875	95.7985	0.03452814	9.871876	2408
245	4	29.5	9.098802	14.20000	74.3000	0.08845976	17.239369	2640
248	6	59.0	4.725800	13.83333	66.7000	0.12963260	10.517350	2129
246	15	19.5	2.638343	13.10000	55.7000	0.08227567	7.310082	2535

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

	chrom	locus	variance	locus.LCL	locus.UCL	variance.LCL	variance.UCL	n.qtl
247	1	69.9	4.768429	43.7	77.60	43.7	77.60	102
245	4	29.5	11.538096	29.5	30.60	29.5	30.60	102
248	6	61.2	5.173255	54.1	66.70	54.1	66.70	102
246	15	17.5	3.183654	13.1	26.45	13.1	26.45	102

3 Model Diagnostics

A number of diagnostic routines have been described in other vignettes for this package. For instance, `qb.scanone` and `qb.scantwo` can be used to identify the strength of main and epistatic QTL. In addition,

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. All these routines have some connection to R/qtl (www.rqtl.org) routines, such as `scanone`, `scantwo` and `fitqtl`.

4 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]]
```

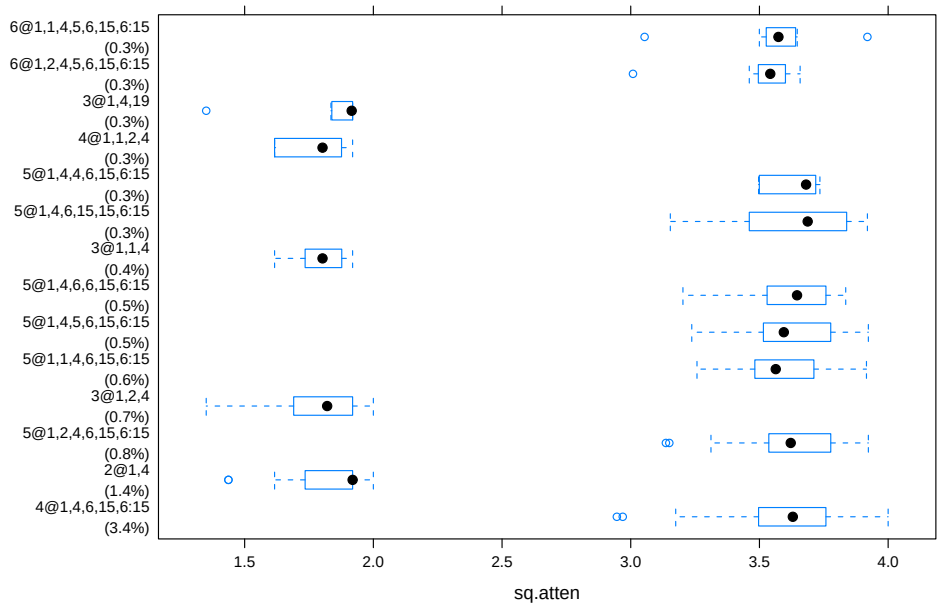
The routine `qb.close` gives a score comparison for each MCMC realization. These are summarized over chromosome pattern, or over number of QTL using boxplots.

```
> 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         3.4 2.946   3.500  3.630 3.613   3.758 4.000
2@1,4                   1.4 1.437   1.735  1.919 1.832   1.919 2.000
5@1,2,4,6,15,6:15       0.8 3.137   3.536  3.622 3.611   3.777 3.923
3@1,2,4                 0.7 1.351   1.700  1.821 1.808   1.919 2.000
5@1,1,4,6,15,6:15       0.6 3.257   3.484  3.563 3.575   3.698 3.916
5@1,4,5,6,15,6:15       0.5 3.237   3.515  3.595 3.622   3.777 3.923
5@1,4,6,6,15,6:15       0.5 3.203   3.541  3.646 3.631   3.757 3.835
3@1,1,4                 0.4 1.616   1.735  1.803 1.790   1.858 1.919
5@1,4,6,15,15,6:15      0.3 3.154   3.461  3.687 3.642   3.839 3.919
5@1,4,4,6,15,6:15       0.3 3.497   3.500  3.681 3.630   3.719 3.735
4@1,1,2,4               0.3 1.616   1.616  1.803 1.775   1.876 1.919
3@1,4,19                0.3 1.351   1.839  1.916 1.837   1.919 1.919
6@1,2,4,5,6,15,6:15     0.3 3.009   3.513  3.542 3.493   3.584 3.658
6@1,1,4,5,6,15,6:15     0.3 3.054   3.540  3.574 3.557   3.638 3.919
```

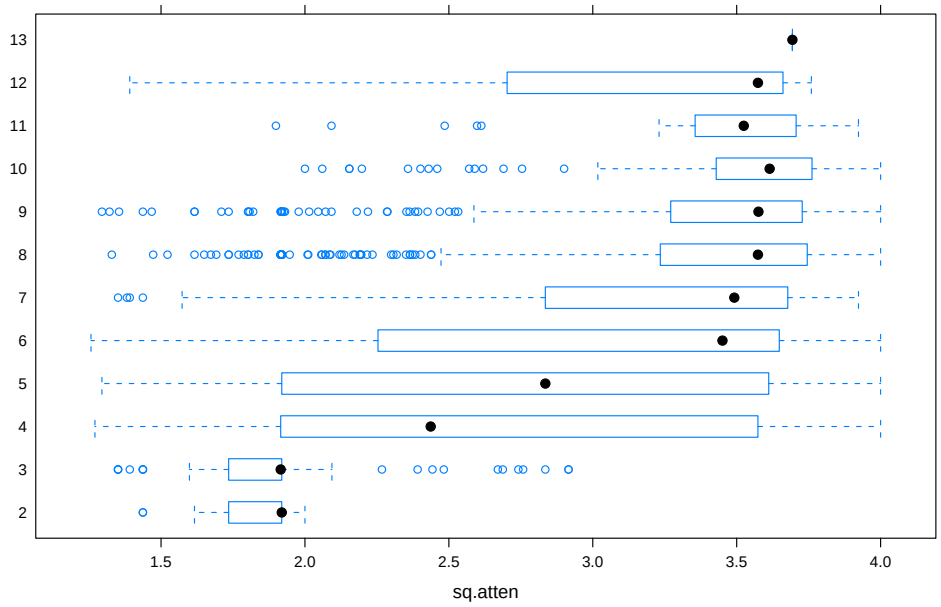
It is more intuitive to look at the boxplots. Notice how patterns that miss the 6:15 interaction have much lower attenuation scores.

```
> 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")
```



5 Multiple Linked Loci

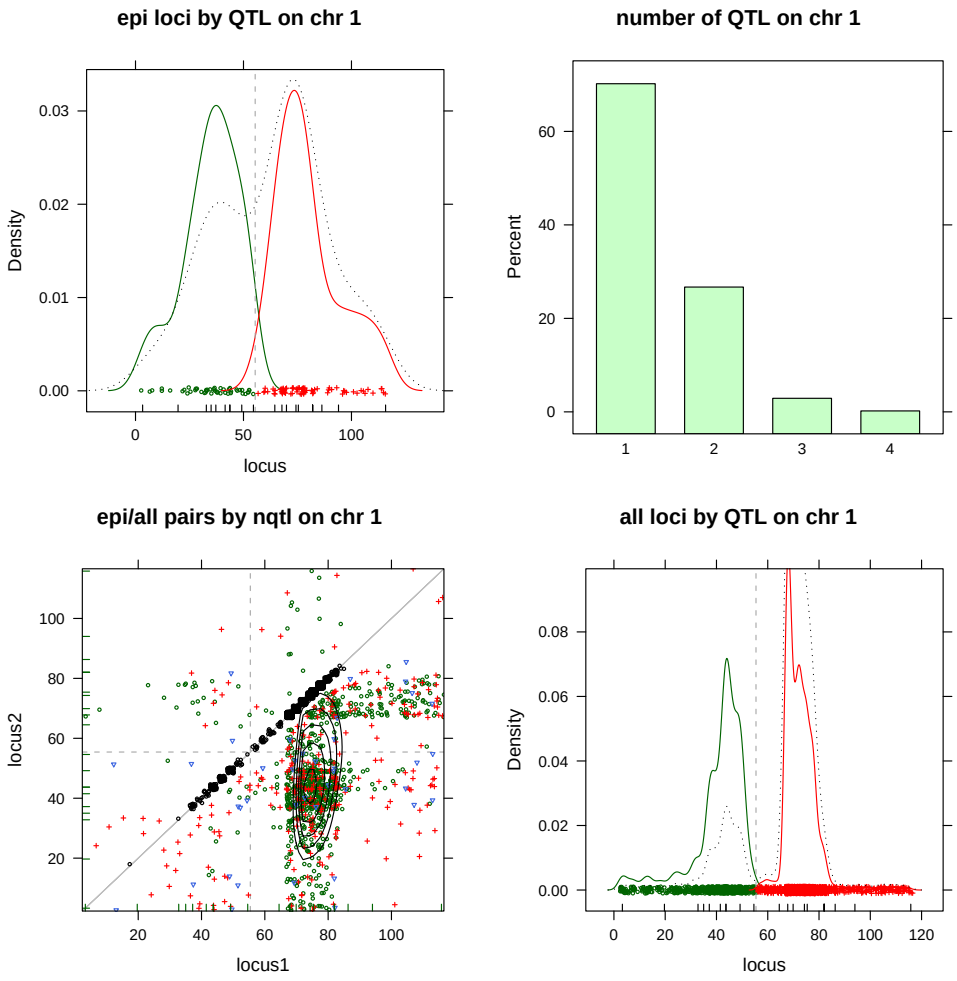
Sometimes there appear to be evidence for linked loci. While 2-dimensional scans with `scantwo` or `qb.scantwo` can disambiguate such situations, it can be helpful to have tools to look finer, and even to

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)
```



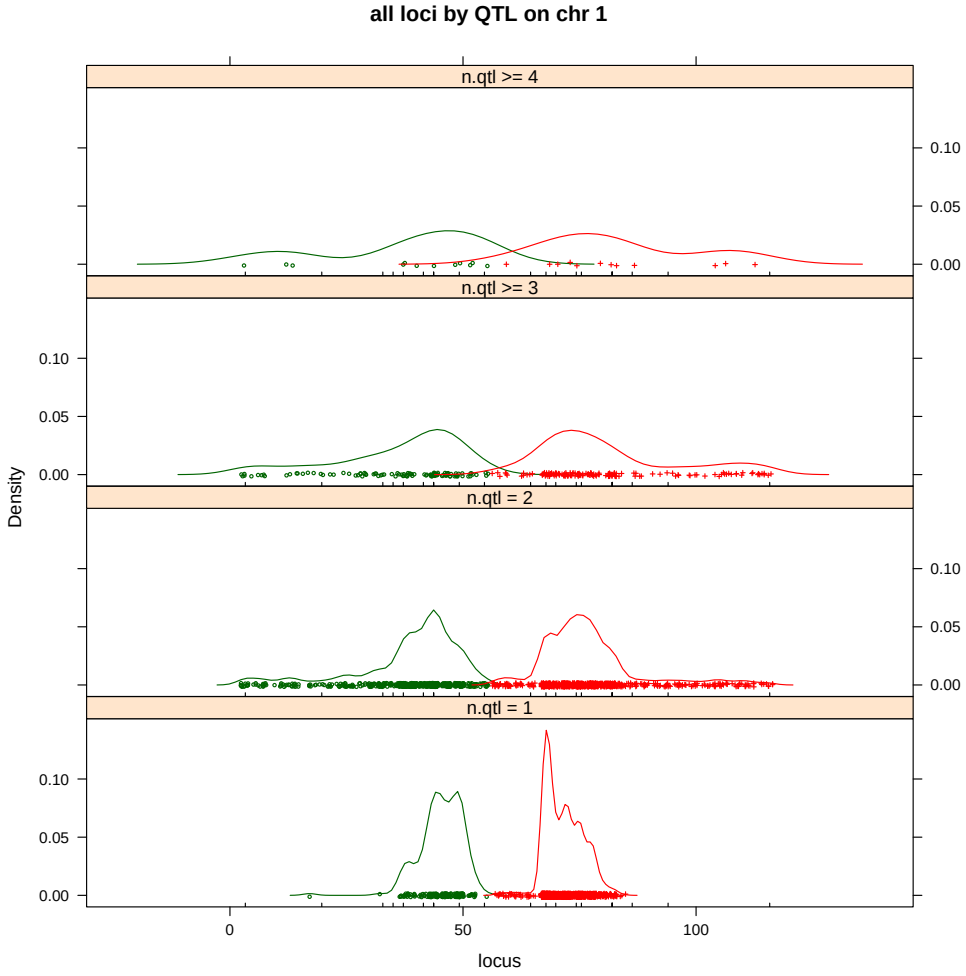

```
> summary(mult)
```

```
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
```

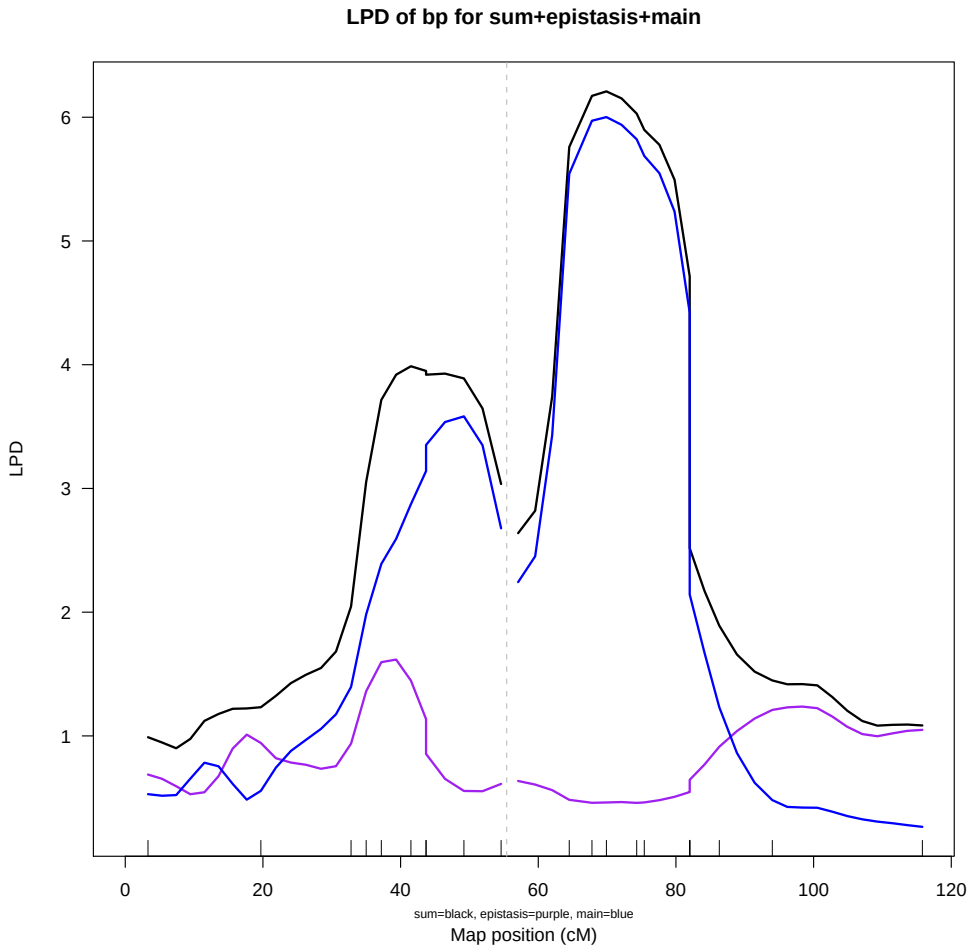
```
> qb.best(qbHyper)
```

```
Maximum number of QTL in architecture: 10

Summary by pattern
                       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

Best pattern(s) by sq.atten score
    chrom locus  variance locus.LCL locus.UCL variance.LCL variance.UCL n.qtl
247   1.2  72.1  4.856429  62.02500     98.36   0.07011681    10.152792  1876
245   4.1  29.5 10.495860  12.17143     37.00   0.16116154    17.797911  1890
248     6  59.0  4.721857  13.83333     66.70   0.14104050    10.436823  1985
246    15  19.5  2.672603  13.10000     55.70   0.08939935     7.274024  2357
```

```
> 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 49.20 49.20 37.20  3.582     1.596  3.889
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 11.347     0.377 11.472
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 13.10 39.57 13.10  0.916     1.853  1.951
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  1.320     6.153  7.112
16  0.0813  8.37  8.37 10.46  0.396     1.710  1.744
17  0.1123 50.30 50.30 50.30  0.377     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)
```


LPD of bp for sum+epistasis+main
LPD
1
2
3
4
5
6
0
20
40
60
80
100
120
sum=black, epistasis=purple, main=blue
Map position (cM)