Prototype QTL Strategy: Phenotype bp in Cross hyper

Brian S. Yandell, W. Whipple Neely, Tapan Mehta, Daniel Schriner, Samprit Banerjee, and Nengjun Yi

May 24, 2006

1 Overview

This document analyzes trait bp for dataset hyper using the 1-D and 2-D Bayesian genome scan routines that build on Markov chain Monte Carlo (MCMC) samples from a posterior for the genetic architecture of a trait. Below the generic `cross` is actually the cross passed by the user in a call to `bmq.sweave`. This entire document was created automatically by a function call in `R`. The function is not yet part of `R/bmqtl`. The actual call was

```
> library(bmqtl)
```

```
> bmq.sweave(hyper, pheno.col = 1,
+  n.iter = 3000, n.draws = 64,
+  threshold = c(main = 3, epistasis = 3, upper = 2),
+  SweaveFile = /u/y/a/yandell/public/statgen/R/bmqtl/doc/hyperpaper.Rnw,
+  PDFDir = bpPDF,
+  remove.bmq = FALSE)
```

At present, the `threshold` values are somewhat arbitrary, chosen for the hyper dataset to pick up apparently real QTL and previously detected epistasis.

This document automates a search for main and epistatic QTL. The main QTL positions are reliably estimated using `bscanone`. This pass also seems to capture the major chromosomes possibly involved in epistasis, although it does not provide very good estimates of positions of purely epistatic QTL within those chromosomes. Next we use `bscantwo` and to identify which pairs of QTL are epistatic, and to get initial estimates of their positions. We refine there positions with `bmq.slice`. Along the way, we use generic `summary` and `plot` routines to view results.

Once we have reasonable estimates of QTL postions and effects, we use confirmatory ANOVA tools to refine the model. That is, we use R/qtl's simulation-based `fitqtl` followed by a stepwise backward fitting approach, using a new `step.fitqtl`, to confirm key QTL. That completes this automated analysis. It would be possible to add other, user-supplied refined analysis at the end of this document if desired.

2 Generating Samples

Here is a summary of the `cross` copy of the hyper object, followed by creation of 3000 MCMC samples.

```
> summary(cross)

    Backcross

    No. individuals:  250

    No. phenotypes:   1
    Percent phenotyped:  100
```

```
    No. chromosomes:  19
    Total markers:    170
    No. markers:      22 8 6 20 14 11 7 6 5 5 14 5 5 5 11 6 12 4 4
    Percent genotyped:  47.9
    Genotypes (%):      AA:50.1  AB:49.9
```

```
> cross <- bmq.genoprob(cross, step = 2)
```

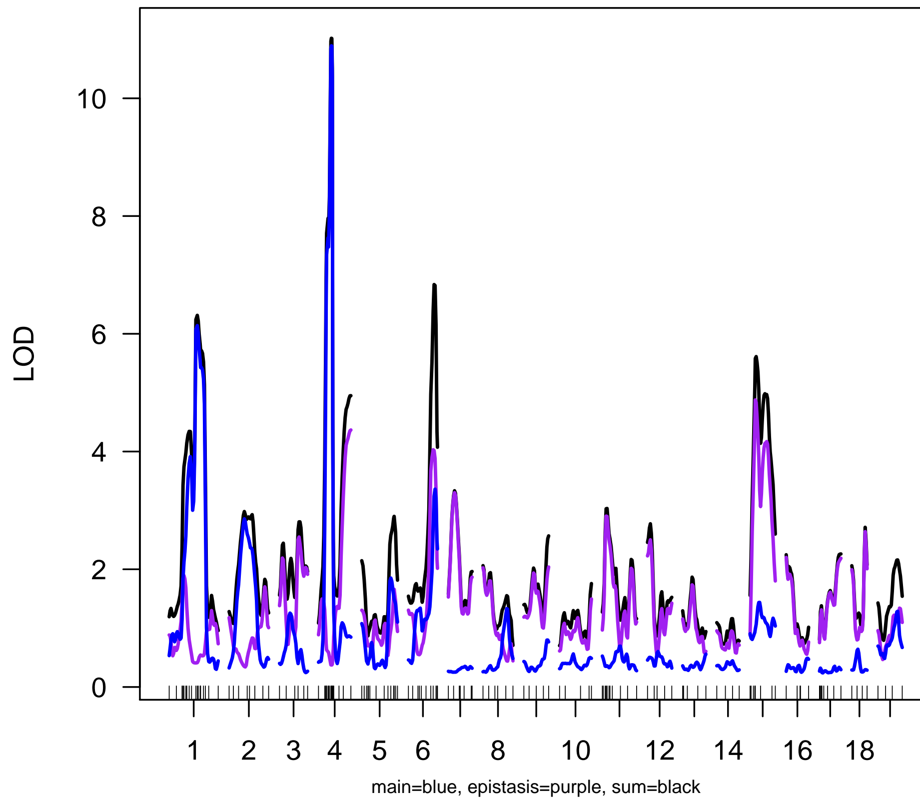
```
> cross.bmq <- bmq.mcmc(cross, genoupdate=TRUE, n.iter = 3000)
```

3 1-D Scans

Now a 1-D scan picks out the major effects. The blue curve represents the LOD score for the `main` effect of a QTL at each locus, while the purple curve shows the LOD score for any `epistasis` between a locus and any other loci. The black curve shows the combination of main and epistatic effects.

```
> one <- bscanone(cross.bmq, type = "LOD")
> plot(one, smooth = 3)
> summary(one)
```

```
   chr    pos   main epistasis    sum niter
1    1 67.800  6.315     2.109  6.462  4054
2    2 42.633  3.098     2.584  3.271  1563
3    3 26.256  1.348     3.380  3.785   513
4    4 29.500 10.983     4.744 11.099  4149
5    5 66.700  1.995     2.214  3.288   833
6    6 61.200  3.851     4.637  7.579  1612
7    7 41.300  0.693     3.446  3.480   583
8    8 61.050  1.678     2.290  2.369   452
9    9 64.867  1.025     2.520  3.292   400
10  10 33.829  1.005     1.867  1.905   341
11  11 43.700  0.856     4.837  5.166   519
12  12 25.175  1.357     3.409  3.443   337
13  13 59.000  0.583     3.326  3.366   290
14  14 50.450  0.766     1.226  1.633   210
15  15 27.500  1.754     5.786  6.361  1957
16  16 29.500  0.662     2.719  2.779   255
17  17 33.900  0.673     2.238  2.351   374
18  18 12.200  1.001     4.001  4.047   208
19  19 47.373  1.719     1.896  3.137   379
```



We can mostly automate the selection of peaks. We are still working on how to set reasonable thresholds, but for now a threshold of 4 for the overall (`sum`) and 2 for epistasis picks up the key features nicely. The variable `threshold` was set in the call to `bmq.sweave` that created this document.

```
> threshold

     main epistasis     upper
        3         3         2

> sum.one <- summary(one, threshold = threshold, sort = "sum")
> sum.one

   chr    pos   main epistasis    sum niter
4    4 29.500 10.983     4.744 11.099  4149
6    6 61.200  3.851     4.637  7.579  1612
1    1 67.800  6.315     2.109  6.462  4054
15  15 27.500  1.754     5.786  6.361  1957
11  11 43.700  0.856     4.837  5.166   519
18  18 12.200  1.001     4.001  4.047   208
3    3 26.256  1.348     3.380  3.785   513
7    7 41.300  0.693     3.446  3.480   583
12  12 25.175  1.357     3.409  3.443   337
13  13 59.000  0.583     3.326  3.366   290
2    2 42.633  3.098     2.584  3.271  1563
```

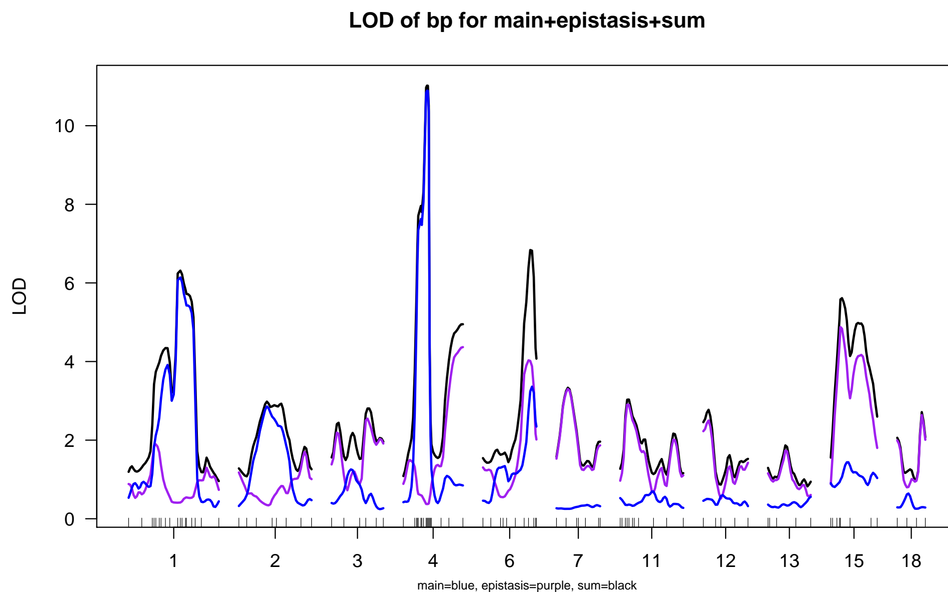
The new variables chr and pos capture the chromosome numbers and main QTL positions, respectively.

```
> chrs <- sort(as.vector(sum.one[, "chr"]))
> pos <- sum.one[as.character(chrs), "pos"]
> pos

     1      2      3      4      6      7     11     12     13     15     18
67.800 42.633 26.256 29.500 61.200 41.300 43.700 25.175 59.000 27.500 12.200
```

The following two figures highlight the selected chromosomes. The first shows the LODs

```
> plot(one, chr = chrs, smooth = 3)
```



and the second shows the marginal means, which are roughly symmetric about the phenotype mean of 101.

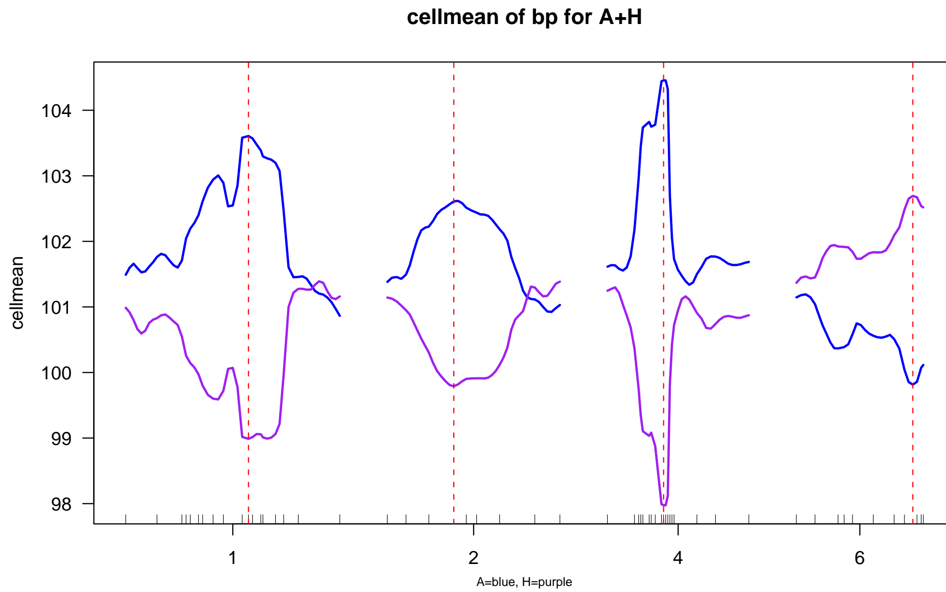
```
> tmp <- sum.one[as.character(chrs), "main"] >= threshold["main"]
> chr1 <- chrs[tmp]
> one <- bscanone(cross.bmq, chr = chr1, type = "cellmean")
> summary(one)

  chr    pos       A       H niter
1   1 66.062 103.076  99.371  4054
2   2 50.705 102.264 100.120  1563
4   4 35.683 103.681  98.805  4149
6   6 49.945 100.241 102.266  1612

> pos1 <- pos[tmp]
> pos.plot <- map.pos(cross, chr1, pos1)
> pos.plot

      1       2       4       6
 64.500 172.433 282.700 413.700

> plot(one, smooth = 3)
> abline(v = pos.plot, lty = 2, col = "red")
```



The variable `pos.plot` is a technical device to plot vertical red lines at the maximum LODs.

4 2-D Scan

Now a look at a 2-D scan reveals the strength of epistasis. The summary suggests some other epistases, but some of this may be spurious [i.e. we will want to look further!]. We consider a subset of this summary above the `upper` threshold of 2.

```
> two <- bscantwo(cross.bmq, chr = chrs, type = "LOD")
> maxpairs
```

```
[1] 20  5
```

```
> sum.two <- summary(two, threshold = threshold, maxpairs = maxpairs[1])
> sum.two
```

```
      chr1 chr2    pos1   pos2  lower upper niter
1.1      1    1  30.617 30.617 12.829 2.618  1182
1.6      1    6 104.900  2.450 12.956 3.157  2115
1.7      1    7  35.000 17.467 16.167 5.228   420
2.3      2    3  60.650 60.100 12.361 3.759   279
3.7      3    7  46.983 53.600 10.792 2.319   324
3.11     3   11  13.129 70.500 12.729 2.764     4
3.13     3   13  30.633 27.309 12.731 3.052   298
3.15     3   15  10.943 63.400 17.063 7.057   366
4.6      4    6  67.737 42.600 12.526 3.414   348
4.15     4   15  74.300 41.592 14.451 5.544   178
4.18     4   18  14.200  2.200 10.742 2.732   159
6.11     6   11  66.700 66.340 10.675 5.306   125
6.15     6   15  51.400 23.500 13.261 3.246   380
7.11     7   11  37.200 19.700 11.551 2.660   421
11.11   11   11  33.367 64.260 14.949 5.577   814
11.12   11   12  29.233 48.600 11.792 3.678   536
12.13   12   13  18.600 50.733  9.382 2.276   343
```

Now let's extract the genetic architecture implied by this evidence for epistasis. The `arch$pair` provides information on what loci pairs are epistatic, while `archpairs` shows what chromosomes are involved in these epistatic pairs. Some effort is made to merge nearby QTL estimated positions in `bmq.mergeqtl`.

```
> arch <- bmq.mergeqtl(chrs, pos, sum.two)
> t(arch$qtl)

        1    2     3     4     5     6     7     8    9   10   11    12   13
chr  1.00  1.0   1.0  2.00  2.00  3.00  3.00  3.00  3.0  4.0  4.0  4.00 6.00
pos 32.08 67.8 104.9 42.63 60.65 12.04 28.44 46.98 60.1 14.2 29.5 71.02 2.45
       14    15    16   17    18   19    20    21   22    23    24   25    26
chr  6.00  7.00  7.00  7.0 11.00 11.0 11.00 12.00 12.0 13.00 13.00 15.0 15.00
pos 55.47 17.47 39.25 53.6 27.43 43.7 67.03 21.89 48.6 27.31 54.87 25.5 41.59
      27   28
chr 15.0 18.0
pos 63.4  7.2

> if (!is.null(arch$pairs)) t(arch$pairs)

    1  2 3  4  5  6  7  8  9 10 11 12 13 14 15 16
q1  3  1 5  8  6  7  6 12 12 10 14 14 16 18 18 21
q2 13 15 9 17 20 23 27 14 26 28 20 25 18 20 22 24

> archpairs <- bmq.archpairs(arch)
> if (!is.null(archpairs)) t(archpairs$chr)

   1 2 3 4  5  6  7 8  9 10 11 12 13 14 15 16
q1 1 1 2 3  3  3  3 4  4  4  6  6  7 11 11 12
q2 6 7 3 7 11 13 15 6 15 18 11 15 11 11 12 13
```

5 ANOVA Model Fit

Here we want to merge the 1-D `chrs` and `pos` with the 2-D epistatic pairs to determine the chromosomes and positions to include in an ANOVA fit. We equate QTL that are within, say 10cM of each other. After fitting a (very) full model, we use `step.fitqtl`, a newly written routine, to step-by-step reduce the model to key main effects and interactions, preserving hierarchy.

Here are the full set of QTL to be considered. The `pairs` are indexed into the `chr` and `pos`.

```
> arch <- bmq.mergeqtl(chrs, pos, sum.two)
> arch

$qtl
   chr    pos
1    1  32.08
2    1  67.80
3    1 104.90
4    2  42.63
5    2  60.65
6    3  12.04
7    3  28.44
8    3  46.98
9    3  60.10
10   4  14.20
11   4  29.50
12   4  71.02
```

```
13   6   2.45
14   6  55.47
15   7  17.47
16   7  39.25
17   7  53.60
18  11  27.43
19  11  43.70
20  11  67.03
21  12  21.89
22  12  48.60
23  13  27.31
24  13  54.87
25  15  25.50
26  15  41.59
27  15  63.40
28  18   7.20

$pairs
   q1 q2
1   3 13
2   1 15
3   5  9
4   8 17
5   6 20
6   7 23
7   6 27
8  12 14
9  12 26
10 10 28
11 14 20
12 14 25
13 16 18
14 18 20
15 18 22
16 21 24
```

The following uses R/qtl tools `calc.genoprob`, `sim.geno` and `makeqtl`, plus the new `step.fitqtl`, which calls `fitqtl` multiple times.

```
> cross <- calc.genoprob(clean(cross), step = 2, error = 0.01)
> n.draws

[1] 64

> cross <- sim.geno(cross, n.draws = n.draws, step = 2, error = 0.01)
> qtl <- makeqtl(cross, arch$qtl$chr, arch$qtl$pos)
> cross.step <- step.fitqtl(cross, qtl, pheno.col, arch)

   drop                     LOD     p
1  Chr11@43.7               -0.0714 1.0000
2  Chr3@46.98:Chr7@53.6     -0.0117 1.0000
3  Chr4@14.2:Chr18@7.2      -0.0243 1.0000
4  Chr3@12.04:Chr11@67.03    0.0187 0.7890
5  Chr18@7.2                 0.1650 0.4260
6  Chr4@14.2                 0.3800 0.2260
```

```
7  Chr7@53.6                  0.3670 0.2330
8  Chr2@42.63                 0.3230 0.2620
9  Chr6@55.47:Chr11@67.03     0.3210 0.2620
10 Chr3@46.98                 0.3850 0.2190
11 Chr12@21.89:Chr13@54.87    0.4840 0.1670
12 Chr13@54.87                0.4150 0.1990
13 Chr12@21.89                0.4010 0.2060
14 Chr3@28.44:Chr13@27.31     0.5290 0.1460
15 Chr13@27.31                0.6390 0.1090
16 Chr7@39.25:Chr11@27.43     0.8280 0.0672
17 Chr7@39.25                 0.6220 0.1120
18 Chr4@71.02:Chr6@55.47      0.8390 0.0643
19 Chr3@12.04:Chr15@63.4      1.1000 0.0337
20 Chr3@12.04                 0.7100 0.0872
21 Chr15@63.4                 0.6910 0.0910
22 Chr3@28.44                 0.6660 0.0962
23 Chr11@27.43:Chr11@67.03    1.0700 0.0343
24 Chr11@67.03                0.5270 0.1370
25 Chr11@27.43:Chr12@48.6     1.3000 0.0194
26 Chr12@48.6                 0.6480 0.0978
27 Chr11@27.43                1.0700 0.0334
28 Chr2@60.65:Chr3@60.1       0.9960 0.0393
29 Chr3@60.1                  0.0997 0.5130
30 Chr1@104.9:Chr6@2.45       1.1600 0.0254
31 Chr6@2.45                  0.0200 0.7690
32 Chr1@104.9                 0.4020 0.1870
33 Chr2@60.65                 1.4300 0.0125
```

```
> cross <- clean(cross)
```

Now we run stepwise backward elimination, preserving hierarchy.

```
> cross.step <- step.fitqtl(cross, qtl, pheno.col, arch)
```

```
   drop                       LOD     p
1  Chr11@43.7                -0.0714 1.0000
2  Chr3@46.98:Chr7@53.6      -0.0117 1.0000
3  Chr4@14.2:Chr18@7.2       -0.0243 1.0000
4  Chr3@12.04:Chr11@67.03     0.0187 0.7890
5  Chr18@7.2                  0.1650 0.4260
6  Chr4@14.2                  0.3800 0.2260
7  Chr7@53.6                  0.3670 0.2330
8  Chr2@42.63                 0.3230 0.2620
9  Chr6@55.47:Chr11@67.03     0.3210 0.2620
10 Chr3@46.98                 0.3850 0.2190
11 Chr12@21.89:Chr13@54.87    0.4840 0.1670
12 Chr13@54.87                0.4150 0.1990
13 Chr12@21.89                0.4010 0.2060
14 Chr3@28.44:Chr13@27.31     0.5290 0.1460
15 Chr13@27.31                0.6390 0.1090
16 Chr7@39.25:Chr11@27.43     0.8280 0.0672
17 Chr7@39.25                 0.6220 0.1120
18 Chr4@71.02:Chr6@55.47      0.8390 0.0643
19 Chr3@12.04:Chr15@63.4      1.1000 0.0337
20 Chr3@12.04                 0.7100 0.0872
```

```
21 Chr15@63.4              0.6910 0.0910
22 Chr3@28.44              0.6660 0.0962
23 Chr11@27.43:Chr11@67.03 1.0700 0.0343
24 Chr11@67.03             0.5270 0.1370
25 Chr11@27.43:Chr12@48.6  1.3000 0.0194
26 Chr12@48.6              0.6480 0.0978
27 Chr11@27.43             1.0700 0.0334
28 Chr2@60.65:Chr3@60.1    0.9960 0.0393
29 Chr3@60.1               0.0997 0.5130
30 Chr1@104.9:Chr6@2.45    1.1600 0.0254
31 Chr6@2.45               0.0200 0.7690
32 Chr1@104.9              0.4020 0.1870
33 Chr2@60.65              1.4300 0.0125
```

```
> summary(cross.step$fit)

                Summary for fit QTL

Method is:  imp
Number of observations:  250

Full model result
----------------------------------
Model formula is:  y ~ Q1 + Q2 + Q11 + Q12 + Q14 + Q15 + Q25 + Q26 + Q1:Q15 + Q12:Q26 +

 Model formula is:      Q14:Q25

       df        SS        MS      LOD     %var Pvalue(Chi2) Pvalue(F)
Model  11  6633.612 603.05568 25.55317 37.54393            0         0
Error 238 11035.324  46.36691
Total 249 17668.936


Drop one QTL at a time ANOVA table:
----------------------------------
                       df Type III SS      LOD     %var  F value Pvalue(Chi2)
Chr1@32.08              2     577.825    2.771    3.270    6.231        0.002
Chr1@67.8               1     889.863    4.210    5.036   19.192     1.07e-05
Chr4@29.5               1    2545.933   11.269   14.409   54.908     5.85e-13
Chr4@71.02              2     821.175    3.896    4.648    8.855     1.27e-04
Chr6@55.47              2    1094.384    5.133    6.194   11.801     7.36e-06
Chr7@17.47              2     485.099    2.335    2.745    5.231        0.005
Chr15@25.5              2     910.625    4.304    5.154    9.820     4.96e-05
Chr15@41.59             2     680.239    3.247    3.850    7.335        0.001
Chr1@32.08:Chr7@17.47   1     444.197    2.142    2.514    9.580        0.002
Chr4@71.02:Chr15@41.59  1     642.800    3.073    3.638   13.863     1.68e-04
Chr6@55.47:Chr15@25.5   1     689.513    3.290    3.902   14.871     9.92e-05
                       Pvalue(F)
Chr1@32.08              0.002303 **
Chr1@67.8               1.77e-05 ***
Chr4@29.5               2.19e-12 ***
Chr4@71.02              0.000195 ***
Chr6@55.47              1.30e-05 ***
Chr7@17.47              0.005980 **
Chr15@25.5              7.98e-05 ***
```

```
Chr15@41.59             0.000810 ***
Chr1@32.08:Chr7@17.47   0.002203 **
Chr4@71.02:Chr15@41.59  0.000245 ***
Chr6@55.47:Chr15@25.5   0.000148 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The final model may be more complicated than a model found 'by hand' using standard R/qtl tools. Hopefully that model is a subset of the automatically found model.

6 2-D Epistasis Plots

Should there be any evidence for epistasis that is confirmed by ANOVA, it can be useful to view 2-D plots similar to `scantwo`, but now using the marginal LOD. For technical reasons, we need the `bmq.genoprob` annotations again.

```
> cross <- bmq.genoprob(cross, step = 2)
```

Here is the reduced genetic architecture:

```
> arch2 <- cross.step$arch
> t(arch2$qtl)
```

```
        1    2   11    12    14    15   25    26
chr  1.00  1.0  4.0  4.00  6.00  7.00 15.0 15.00
pos 32.08 67.8 29.5 71.02 55.47 17.47 25.5 41.59
```

```
> if (!is.null(arch2$pairs)) t(arch2$pairs)
```

```
    1  2  3
q1  1 12 14
q2 15 26 25
```

```
> archpairs <- bmq.archpairs(arch2)
> if (!is.null(archpairs)) t(archpairs$chr)
```

```
   1  2  3
q1 1  4  6
q2 7 15 15
```

And here are cliques of chromosomes that are connected through at least one epistatic pair:

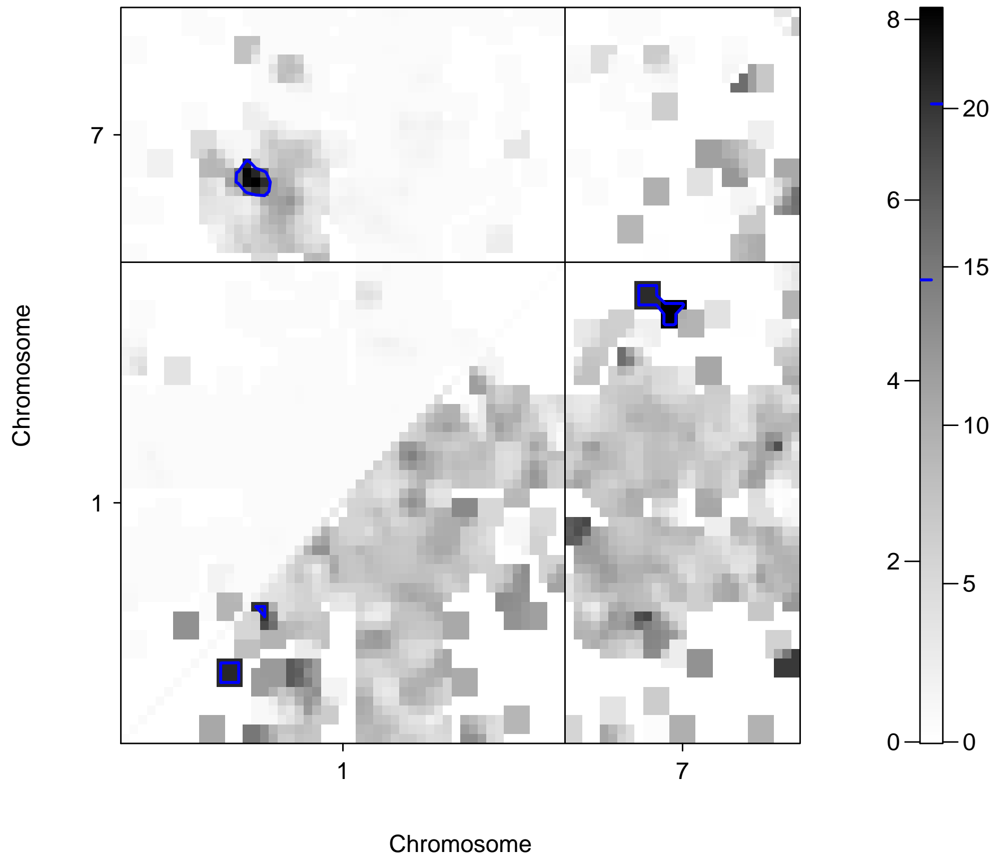
```
> chr2 <- bmq.pairgroup(arch2)
> chr2
```

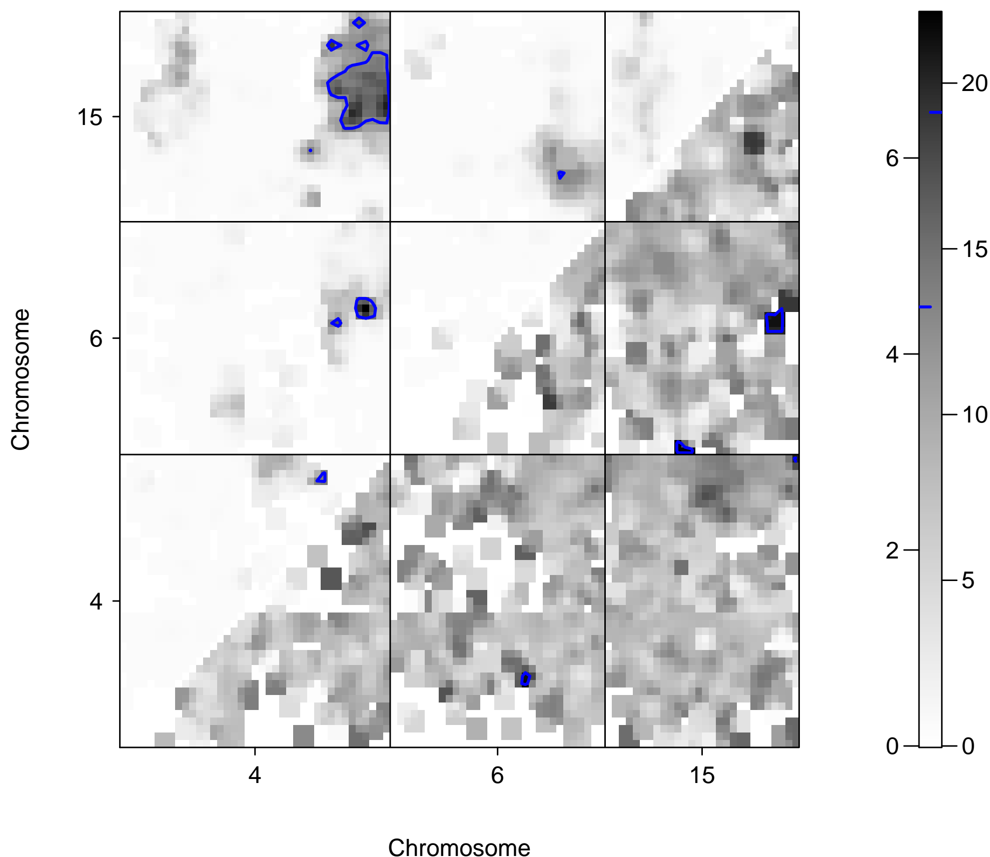
```
[[1]]
[1] 1 7

[[2]]
[1]  4  6 15
```

Here are the plots by clique (if any).

```
> if(length(chr2)) {
+   for(i in seq(length(chr2))
+     plot(two, chr = chr2[[i]], smooth = 3,
+       col = "gray", contour = 3)
+ }
```

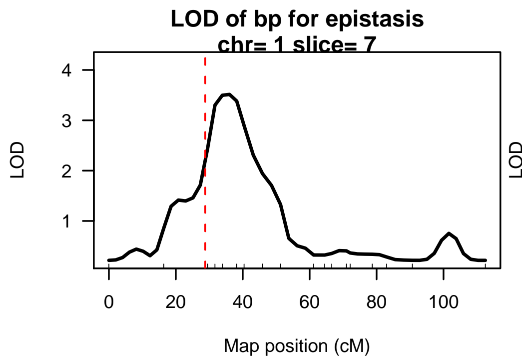

LOD of epistasis / LOD of joint
Chromosome
Chromosome
7
1
1
7
8
6
4
2
0
20
15
10
5
0

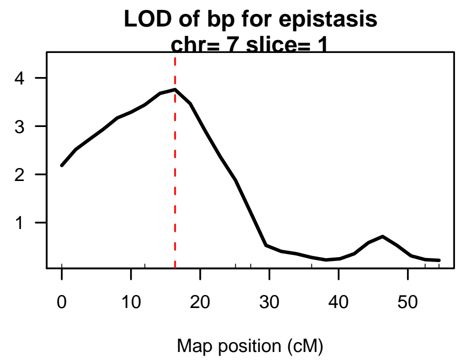


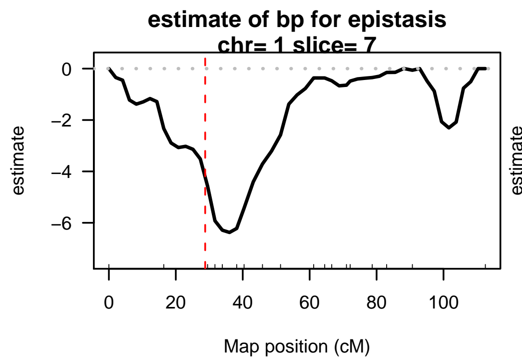
7 1-D Epistasis Slices

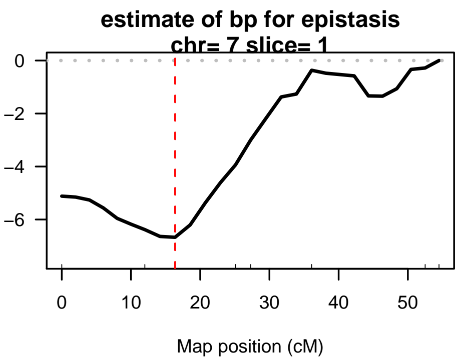
We then examine 1-D slices through the 2-D surface for epistatic pairs in the reduced model, to focus on epistasis for those identified pairs. We show 1-D slices of the LOD and the epistatic effects. In addition, we show interaction plots at the nearest markers.

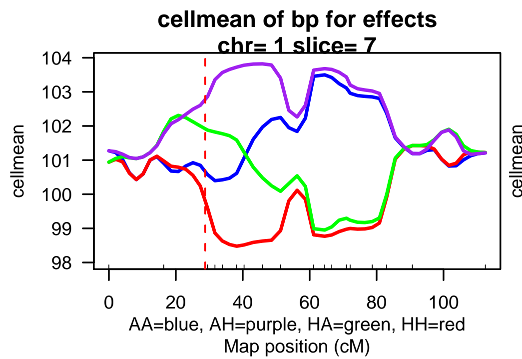
```
> cross <- bmq.genoprob(cross, step=2)
> if(!is.null(archpairs)) {
+  for(i in seq(nrow(archpairs$chr))) {
+    chri <- archpairs$chr[i,]
+    posi <- archpairs$pos[i,]
+    bmq.showtwo(cross.bmq, chri, posi, byrow = FALSE)
+  }
+}
```

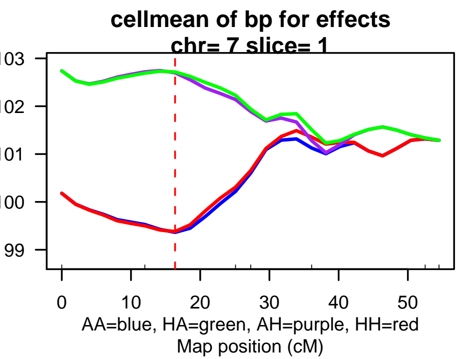

LOD of bp for epistasis
chr= 1 slice= 7
LOD
Map position (cM)

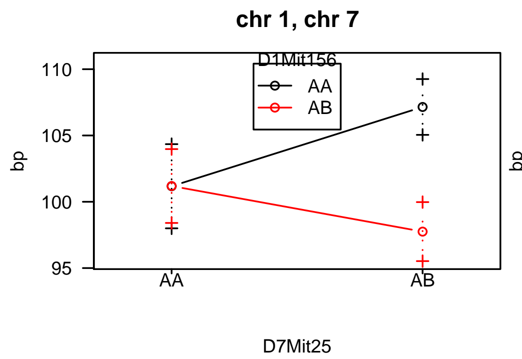

LOD of bp for epistasis
chr= 7 slice= 1
LOD
Map position (cM)

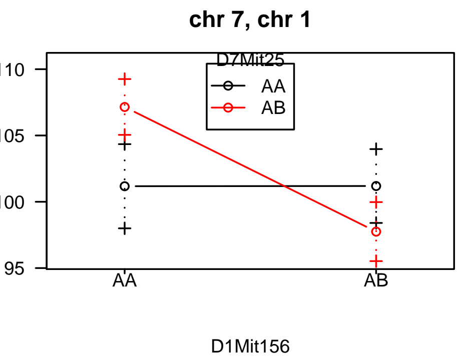

estimate of bp for epistasis
chr= 1 slice= 7
estimate
Map position (cM)

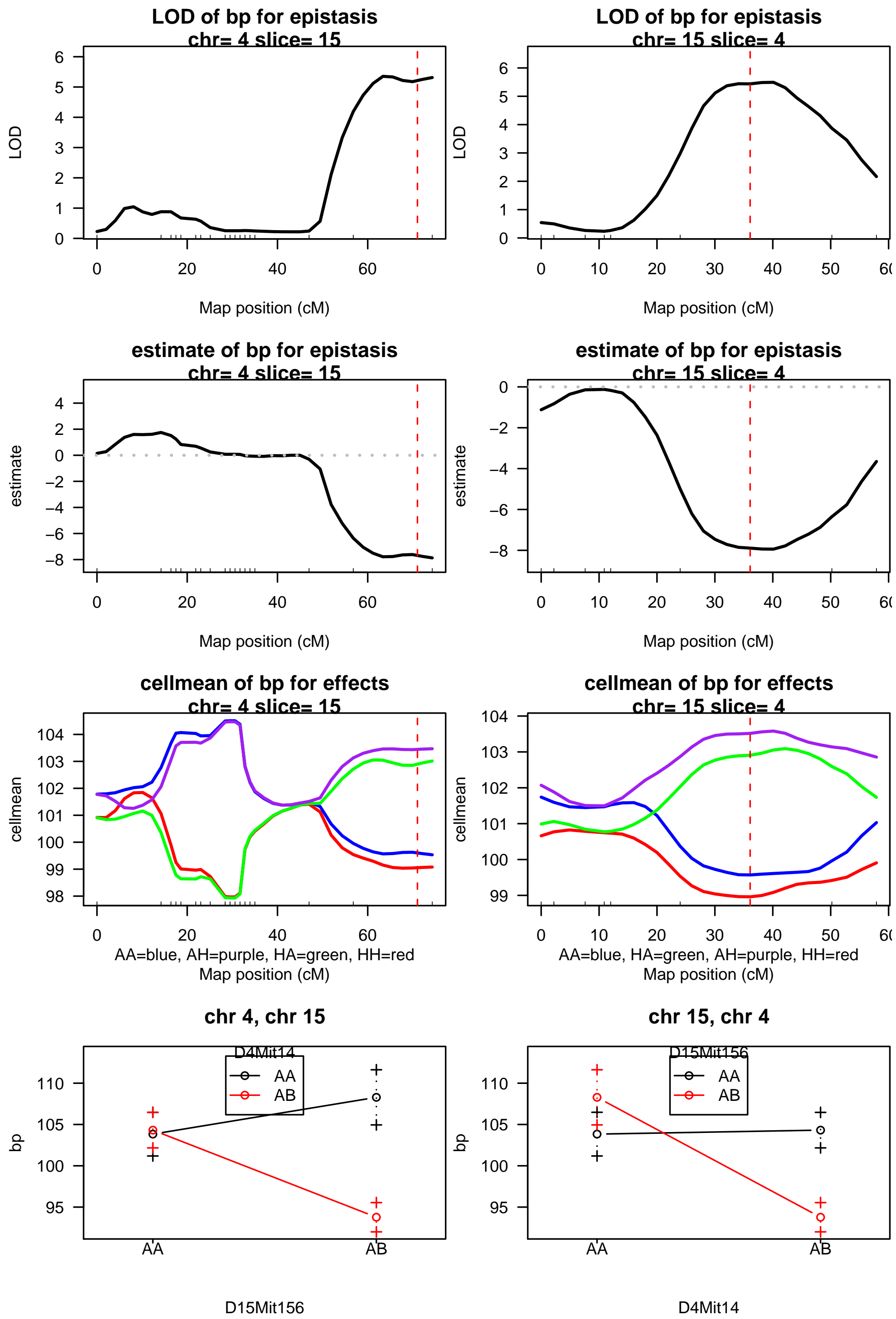

estimate of bp for epistasis
chr= 7 slice= 1
estimate
Map position (cM)

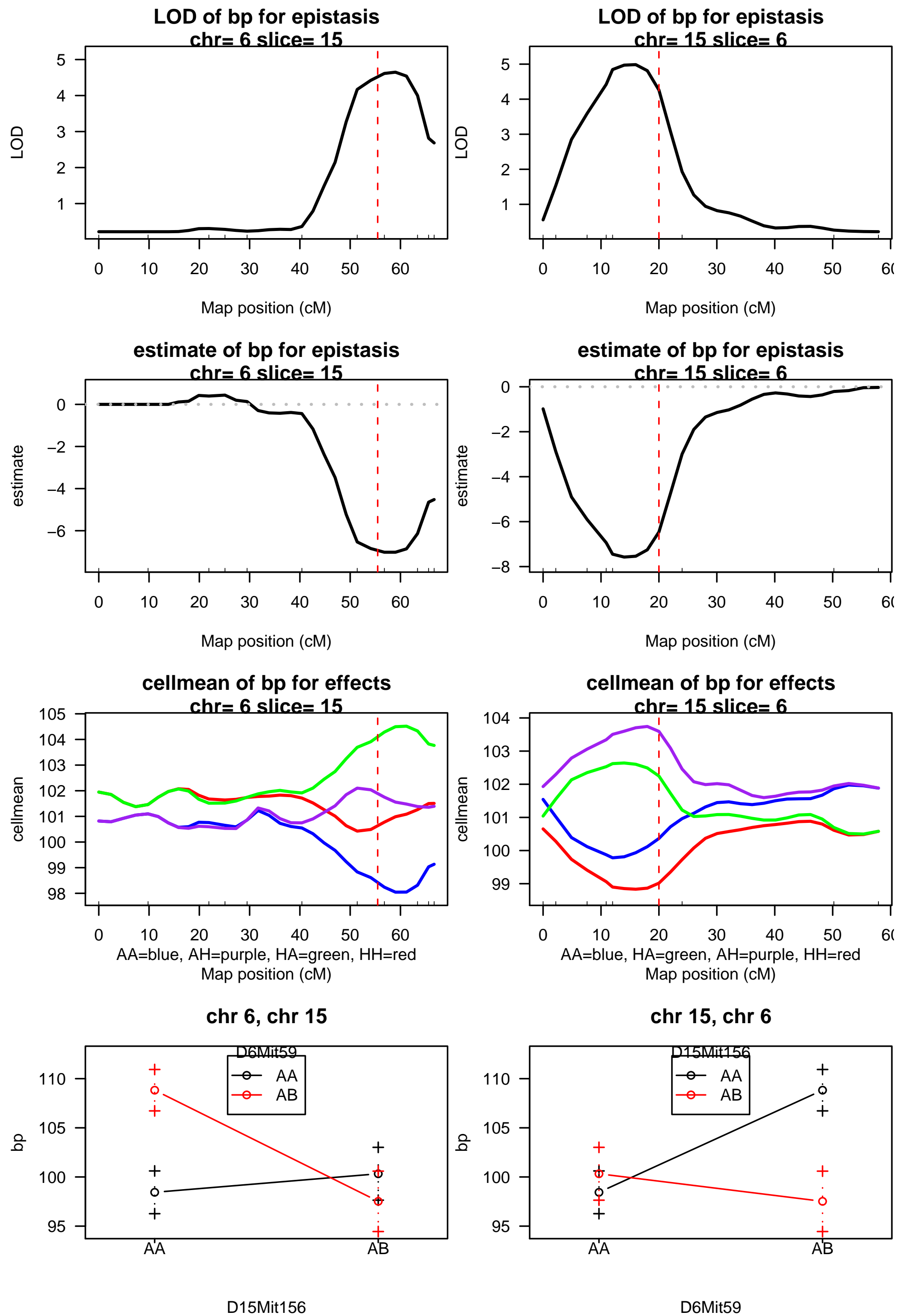

cellmean of bp for effects
chr= 1 slice= 7
cellmean
AA=blue, AH=purple, HA=green, HH=red
Map position (cM)


cellmean of bp for effects
chr= 7 slice= 1
cellmean
AA=blue, HA=green, AH=purple, HH=red
Map position (cM)


chr 1, chr 7
D1Mit156
AA
AB
bp
D7Mit25


chr 7, chr 1
D7Mit25
AA
AB
bp
D1Mit156


LOD of bp for epistasis
chr= 4 slice= 15
LOD
Map position (cM)
LOD of bp for epistasis
chr= 15 slice= 4
LOD
Map position (cM)
estimate of bp for epistasis
chr= 4 slice= 15
estimate
Map position (cM)
estimate of bp for epistasis
chr= 15 slice= 4
estimate
Map position (cM)
cellmean of bp for effects
chr= 4 slice= 15
cellmean
AA=blue, AH=purple, HA=green, HH=red
Map position (cM)
cellmean of bp for effects
chr= 15 slice= 4
cellmean
AA=blue, HA=green, AH=purple, HH=red
Map position (cM)
chr 4, chr 15
D4Mit14
AA
AB
bp
D15Mit156
chr 15, chr 4
D15Mit156
AA
AB
bp
D4Mit14



8 User Customized Section

We know from previous work that there are main QTLs on chromosomes 1 and 4, and epistatic pairs involving 6 and 15, and 7 and 15. Here we pick the nested model that contains these QTL.

```
> arch3 <- bmq.subarch(cross.step, main = c(1, 4), epistasis = data.frame(q1 = c(6,
+     7), q2 = rep(15, 2)))
> t(arch3$qtl)

       2   11    14   25
chr  1.0  4.0  6.00 15.0
pos 67.8 29.5 55.47 25.5

> t(arch3$pairs)

    3
q1 14
q2 25

> cross.step2 <- step.fitqtl(cross, qtl, pheno.col, arch3)
> summary(cross.step2)

     Length Class  Mode
fit  2      fitqtl list
arch 2      -none- list
```

Now we do a formal comparison of this reduced model with the fuller model we automatically uncovered. It appears that the fuller model is a much better fit.

```
> anova.fitqtl(cross.step, cross.step2)

       df        SS        MS      LOD     %var Pvalue(Chi2) Pvalue(F)
Model   6  1459.469 243.24491 6.742999 8.260087     2.06e-05  4.28e-05
Error 238 11035.324  46.36691
Total   0     0.000
```

9 Cleaning Up

That completes the template. Now the penultimate task is to remove the objects created by `R/bmqtl`, if this is desired by the user.

```
> if (remove.bmq) {
+     bmq.remove(cross.bmq)
+     rm(cross, pheno.col, threshold, maxpairs, n.iter, n.draws,
+         remove.bmq)
+ }
```

Finally, run `pdflatex` twice on the file `hyperpaper.tex` and use a free Acrobat reader to view.