Published in final edited form as:

Methods Mol Biol. 2012 ; 871: 205–225. doi:10.1007/978-1-61779-785-9_11.

Identifying QTL Associated with Multiple Traits in Experimental Crosses

Samprit Banerjee, PhD¹ and Nengjun Yi, Ph.D²

¹Division of Biostatistics and Epidemiology, Department of Public Health, Weill Cornell Medical College, 402 E. 67th St, New York, NY 10065

²Section of Statistical Genetics, Department of Biostatistics, University of Alabama, Birmingham, Ryals Public Health Bldg, 317F, University of Alabama at Birmingham, Birmingham, AL 35294

1. Introduction

The broad goal of mapping Quantitative Trait Loci (QTL) associated with complex traits is to shed light on the genetic architecture underlying complex traits. Complex traits are physical expressions of an organism which are controlled by many genes/loci and whose inheritance does not follow simple rules of Mendelian genetics. Over the past few decades there has been considerable development of statistical methodology to map QTL for complex traits. The biggest challenge in this respect is the vast number of unknown genes, their interaction with themselves and with environmental factors. The focus of this chapter is the simultaneous inference of multiple complex traits which has several advantages. First, the chief advantage is in the detection of the phenomenon of pleiotropy, in which action at a single locus leads to variation in a number of traits. For example, one might desire to improve two traits that tend to be negatively correlated. If this negative association is due to pleiotropy, it will be very difficult to improve both traits simultaneously. Second, joint analysis increases the power to detect and the precision of QTL effects. There are several methods for joint analysis, among them, (Jiang and Zeng) and (Banerjee *et al.*) are the most comprehensive in terms of software. In this chapter we will describe the methods developed in (Banerjee *et al.* ; Banerjee *et al.*) and the corresponding freely available software R/qtlbim (Yandell *et al.*).

2. Methods

2.1. Experimental Design

An investigator performing an experiment to identify QTL chooses two pure-breeding lines that differ in the trait of interest (high (H) or low (L) parental strains). These lines are heavily inbred so that within an inbred strain all individuals are essentially genetically identical and are homozygous at all loci. The low parental line is then mated with the high parental line which forms the first filial generation (or F₁). The F₁ generation gets a copy of

---

First Author's Contact Information: Name: Samprit Banerjee, Address: Division of Biostatistics and Epidemiology, Department of Public Health, Weill Cornell Medical College, 402 E. 67th St, New York, NY 10065, Telephone: 646-962-8014, Fax: 646-962-0281, sab2028@med.cornell.edu.

the genetic material from each parental line resulting in heterozygous HL genotypes. The F_1 progenies are then back-mated with the low parental line resulting in backcross progenies which have two possible genotypes HL or LL. The chromosomes cross-over during meiosis and the idea is to capture these cross-overs or recombination events. Another popular design is the intercross or F_2 design where the progenies of F_1 are mated with themselves or each other to generate the F_2 progenies having three possible genotypes HH, LH and LL.

2.2. Genetic Model

In order to discern the association between a locus and a phenotype one needs to develop a statistical model to capture this association. Consider the very simple case of one QTL and one trait resulting in the linear model (1).

$$Y_i = \mu + Q_i + e_i \quad (1)$$

Q_i is the effect of the QTL on the i^{th} individual which can be modeled using the Cockerham's model (Cockerham). The key advantage with Cockerham's model is that it leads to orthogonal contrasts so parameter estimates do not depend on the current complexity of the model allowing testing of non-nested models. For a backcross population with only two segregating genotypes (say, AA and Aa), the contrasts for main effects of $AA(Aa)$ are defined by $1/2$ ($-1/2$). For an F_2 population with three segregating genotypes (say, AA , Aa and aa) the additive contrasts are defined by 1, 0, and -1 while the dominant contrasts are coded as $-1/2$, $1/2$ and $-1/2$ for AA , Aa and aa respectively. The contrasts for dominant effects can be interpreted as deviation from the additive nature of allelic effect, *i.e.* interaction between the two alleles at a particular locus. The epistatic contrasts are defined by multiplying the contrasts of the corresponding main effects. Note, for an F_2 population there can be four different types of epistatic effects between two loci depending on the nature of main effect at each loci; additive-additive, additive-dominant, dominant-additive and dominant-dominant interactions. To construct contrasts for $G \times E$ interaction one can simply multiple the fixed covariate with the corresponding main effect contrast. As an example, the epistatic model for two loci A and B, each with two alleles (A , a , and B , b) can be presented for a F_2 cross in the following way:

$$Q_i = \mu + \beta_1^a x_{1i}^a + \beta_1^d x_{1i}^d + \beta_2^a x_{2i}^a + \beta_2^d x_{2i}^d + \beta_{12}^{aa} x_{1i}^a x_{2i}^a + \beta_{12}^{ad} x_{1i}^a x_{2i}^d + \beta_{12}^{da} x_{1i}^d x_{2i}^a + \beta_{12}^{dd} x_{1i}^d x_{2i}^d \quad (2)$$

by defining the coded variables as

$$x_1^a = \begin{cases} 1 & \text{if A is } AA \\ 0 & \text{if A is } Aa \\ -1 & \text{if A is } aa \end{cases} \quad x_1^d = \begin{cases} 1 & \text{if B is } BB \\ 0 & \text{if B is } Bb \\ -1 & \text{if B is } bb \end{cases}$$

$$x_1^d = \begin{cases} \frac{1}{2} & \text{if A is } Aa \\ -\frac{1}{2} & o.w \end{cases} \quad x_2^d = \begin{cases} \frac{1}{2} & \text{if B is } Bb \\ -\frac{1}{2} & o.w \end{cases}$$

Note that the superscript a and d correspond to the additive and dominant main effects, while aa , ad , da and dd correspond to the additive-additive, additive-dominant, dominant-additive and dominant-dominant interactions.

2.3. Interval Mapping

A QTL associated with a trait might not be located exactly on a particular marker, hence it is important to look for QTL between marker intervals. The idea of interval mapping is to search for candidate QTL between observed marker intervals. This introduces a problem as the genotypes on these putative positions are not observed. One can estimate the genotypes at these candidate loci from the genotypes of the flanking markers and the probability of a recombination event happening between the marker and the putative QTL. For the advanced reader please refer to (Jiang and Zeng) for a detailed description on this process.

2.4. Multivariate Model

The multivariate model is constructed to jointly analyze multiple correlated traits. It is different from a trait-by-trait analysis as it explicitly models the correlation among the traits which is ignored by the trait-by-trait analysis. Let us consider a very simple case of T correlated traits ($\mathbf{y}_t$, $t = 1, 2, \dots, T$) and one QTL Q for a backcross experiment.

$$\mathbf{y}_i = \alpha + \beta \mathbf{Q}_i + \boldsymbol{\varepsilon}_i \quad (3)$$

Here the subscript i corresponds to the i^{th} individual. It is also assumed that $\boldsymbol{\varepsilon}_i \sim N_T(\mathbf{0}, \Sigma)$ i.e. the residuals for i^{th} individual follow a multivariate (T -variate, the subscript of N and note that $\mathbf{0}$ is a T -sized vector of zeroes) normal distribution with a covariance matrix Σ . The trait-by-trait analysis is a special case of multivariate analysis when $\Sigma = I$ (where I is the $T \times T$ identity matrix with 1's in the diagonal and 0's elsewhere) and the traits are assumed to be independent.

A simple multivariate model (3) as proposed in (Jiang and Zeng) answers the question of pleiotropy for a single locus but has several limitations. First, it does not consider the joint effect of several QTL, interactions between these QTL and with environmental factors. Second, it is not practical to assume that the same set of QTL will be associated with all traits. To address these issues the QTL SUR (Seemingly Unrelated Regression) model of (Banerjee *et al.*) and its extension to include gene-gene and gene-environment interaction (Banerjee *et al.*) is presented here. The idea is to fix a total number of detectable QTL per trait to an upper bound L (will be described in more detail in 2.5) and collect the contrasts (described in 2.2) for main-effects (subscript G), gene-gene ($G \times G$) interactions (subscript GG), gene-environment ($G \times E$) interactions (subscript GE) and controlling factors for environmental effects (subscript E) in separate matrices. The resulting model for each trait t is

$$\mathbf{y}_t = \boldsymbol{\mu}_t + \mathbf{X}_E \boldsymbol{\beta}_{tE} + \mathbf{X}_G \boldsymbol{\beta}_{tG} + \mathbf{X}_{GG} \boldsymbol{\beta}_{tGG} + \mathbf{X}_{GE} \boldsymbol{\beta}_{tGE} + \mathbf{e}_t \equiv \boldsymbol{\mu}_t + \mathbf{X}_t \boldsymbol{\beta}_t + \mathbf{e}_t \quad (4)$$

where μ_t is the overall mean for trait t , $\mathbf{X}_{tG}$ is the matrix of the main-effect predictors of L loci, determined from the genotypes as described in 2.2, $\mathbf{X}_{tGG}$, $\mathbf{X}_{tGE}$, $\mathbf{X}_E$, are the design matrices of the epistatic effects, gene-environment interactions and environmental effects; β_{tG} , β_{tGG} , and β_{tGE} are the main effects, G×G interactions or epistatic effects and G×E interactions respectively for the t^{th} trait, β_{tE} , are the effects of the covariates and $\mathbf{e}_t$ is the trait-specific error term. The residual error vector $\mathbf{e}$ ($e_{ti}: t = 1, \dots, T, i = 1, \dots, N$) is assumed to be distributed as $N(\mathbf{0}, \Sigma_{T \times T} \otimes \mathbf{I}_{N \times N})$. Thus, the residual errors are independent among individuals, but are correlated among traits within individuals. Note that design matrices, $\mathbf{X}_{tG}$, $\mathbf{X}_{tGG}$, $\mathbf{X}_{tGE}$, have a subscript t corresponding to the t^{th} trait which allows each trait to have a separate genetic model which main advantage of the QTL SUR model. The special case $\mathbf{X}_{tG} = \mathbf{X}_G$ for all t , $\mathbf{X}_{tGG} = \mathbf{X}_{GG}$ for all t , $\mathbf{X}_{tGE} = \mathbf{X}_{GE}$ for all t is called the “Traditional Multivariate Model” which considers the same genetic model (same set of QTL main, G×G and G×E interaction effects) for all traits.

2.5. Model Selection

The aim of QTL mapping is to identify the number and location of QTL affecting a particular trait. A trick to solve the unknown QTL location problem is to generate a grid of putative QTL positions across the entire genome. The coarseness or fineness of the grid will depend on the marker density and goals of the study in question. Defining such a grid introduces a problem of unknown QTL genotypes which can be estimated using the interval mapping idea of 2.3. These genotypes can be updated in the Bayesian framework to produce robust genotypes. The advanced reader should refer to (Yi ; Yi *et al.*) for details. The problem related to the finding the number and combination of QTL that affects the trait is one of model selection (Broman and Speed). The actual number of detectable QTL for each trait is usually not too large. One can impose an upper bound, L , on the total number of detectable QTL. The upper bound L is larger than the number of detectable QTL with high probability for a given data set and can be set based on the initial analyses using conventional mapping methods on each trait (Yi ; Yi *et al.*). One can assign a vector of unobserved indicator variables $\gamma_t = \{\gamma_{t1}, \dots, \gamma_{tL}\}$ for each trait t to indicate which QTL are included in ($\gamma_{tj} = 1, j = 1, \dots, L$) or excluded from ($\gamma_{tj} = 0, j = 1, \dots, L$) the model. The positions of these L QTL are stored in the vector $\lambda_t = \{\lambda_{t1}, \dots, \lambda_{tL}\}$ for each trait t . Jointly, the vector (λ_t, γ_t) denotes the genetic architecture of the t^{th} trait, *i.e.*, the actual number of QTL ($\sum_j \gamma_{tj}$), their positions and the activity of the associated genetic effect. To construct the linear model, invoke the QTL SUR model of 2.4. Two separate SUR models can be considered. The first model is the one that is described above and is named SURd (SUR modeling with different loci used for all traits) and the second is SURs (SUR modeling with same loci used for all traits) which assumes the same location for all QTL, *i.e.*, $\lambda_1 = \lambda_2 = \dots = \lambda_T \equiv \lambda$ and hence same design matrices $\mathbf{X}_1 = \mathbf{X}_2 = \dots = \mathbf{X}_T \equiv \mathbf{X}$. The design matrix ($\mathbf{X}$) is denoted generically to indicate that the assumptions hold for $\mathbf{X}_G$, $\mathbf{X}_{GG}$ and $\mathbf{X}_{GE}$. It is important to note that the indicator variables (γ_t) are different for each trait, thereby pertaining to the definition of a SUR model. The two common models: the univariate single trait approach (STA) and the traditional multivariate model are special cases under the conditions $\Sigma = \mathbf{I}$ and $\gamma_1 = \gamma_2 = \dots = \gamma_T$. (Banerjee *et al.*) provided evidence in favor of the superior performance of the QTL SUR models compared to the traditional multivariate model.

3. Vignette for Statistical Software

In this section simultaneous inference of multiple correlated traits to identify QTL will be discussed using the package R/qtlbim. The package R/qtlbim can be freely downloaded from <http://cran.r-project.org> or www.qtlbim.com. R/qtlbim depends on certain external R packages, namely, R/qtl, R/coda, R/lattice, R/tools and R/MASS. The last three packages are included in the default installation of R. However, one needs to install R/coda (Plummer *et al.*) and R/qtl (Broman *et al.*) to use R/qtlbim. If the packages are not installed R/qtlbim automatically installs them. To invoke the library of functions distributed by R/qtlbim one should do the following:

```
> library(qtlbim)
Loading required package: qtl
Loading required package: lattice
Loading required package: coda
Loading required package: tools
Loading required package: MASS
```

It is recommended to have a system with at least 1GB of RAM and a disk space of at least 10 MB to run R/qtlbim with multiple traits.

3.1. Data Simulation or real data input

R/qtlbim has an inbuilt function `qb.sim.cross` to simulate 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 200 individuals (`n.ind=200`) of “backcross” (`type= "bc"`) mating design. These individuals are genotyped for 11 equally spaced markers (`n.mar=11`, `eq.spacing=TRUE`) on 19 chromosomes each of length 100 cM (`len=rep(100,19)`). Multiple correlated traits can be generated by specifying the option `multiple.trait=TRUE`. Here two correlated traits (`n.pheno=2`) are generated with a correlation (`sigma = rbind(c(1,0.8),c(0.8,1))`) of 0.8 between the traits. There are 6 QTLs, two on chromosome 1 and 4 and one each on chromosomes 2 and 3. The option `qtl.main` and `qtl.epis` can be used to specify the genetic model, *i.e.* which QTL influences the trait and the size of their effect. The simulation allows the user to specify different or same genetic model for each trait thereby generating data for a Seemingly Unrelated Regression model or a Traditional Multivariate Model (as described in 2.4) respectively. Two covariates are generated: a fixed binary covariate with an effect size of 0.5 and a random covariate of five levels whose coefficient are drawn from $N(0,0.07)$. Gene-by-environment interaction (option `gbye`) are specified with a fixed covariate. 3% of the genotypes (`missing.geno=0.03`) and 5% of the phenotypes (`missing.pheno=0.05`) were assumed to be missing randomly.

```
> cross <- qb.sim.cross(len = rep(100,19),n.mar = 11, eq.spacing =TRUE,
+   n.ind = 200, type="bc", missing.geno = 0.05, missing.pheno = 0.03,
```

```

+   multiple.trait=TRUE,n.pheno=2,sigma=rbind(c(1,0.8),c(0.8,1)),
+   qtl.pos=rbind(qtl.1=c(chr=1,pos=15),
+                 qtl.2=c(1,65),
+                 qtl.3=c(2,12),
+                 qtl.4=c(3,15),
+                 qtl.5=c(4,23),
+                 qtl.6=c(4,65) ),
+   qtl.main=rbind(main.1=c(ph=1,qtl=1,add=1.5),
+                 main.2=c(1,2,0.7),
+                 main.3=c(1,3,1),
+                 main.4=c(2,3,1),
+                 main4=c(2,4,0.7),
+                 main5=c(2,6,1.2) ),
+   qtl.epis=rbind(epis1=c(ph=1,qtl.a=2,qtl.b=3,aa=-2),
+                 epis2=c(ph=1,1,4,2),
+                 epis3=c(ph=2,2,4,2),
+                 epis4=c(ph=2,1,2,-2)),
+   gbye = rbind(GxE.1=c(ph=1,qtl=1,add=1.5),
+               GxE.2=c(ph=1,qtl=3,add=-1.5),
+               GxE.3=c(ph=2,qtl=3,add=-1.5),
+               GxE.4=c(ph=2,qtl=6,add=1.5)),
+   covariate = c(fix.cov=0.5,ran.cov=0.07))

```

The function `qb.sim.cross` generates an object of class “`cross`”. In the case of real datasets, the function `read.cross` of `R/qtl` can be used to read datasets in a variety of formats and store it as a “`cross`” object. The simulated cross object has an extra element named “`qtl`” which stores the true QTL positions, genotypes and heritability of effects for post-analysis comparison. The summary of the same can be viewed as shown below and one can get the true parameter values.

```

> names(cross)
[1] "geno" "pheno" "qtl"
> summary(cross$qtl)
$pos
      chr  pos
qtl.1   1   15
qtl.2   1   65
qtl.3   2   12
qtl.4   3   15
qtl.5   4   23
qtl.6   4   65
$herit.main
      pheno qtl      add
main.1     1  1 0.07762380

```

```

main.2      1  2  0.06415190
main.3      1  3  0.06415190
main.4      2  3  0.07837382
main.5      2  4  0.07837382
main.6      2  6  0.11285830
$herit.epis
      pheno  qtl.a  qtl.b      aa
epis.1      1      2      3 0.06415190
epis.2      1      1      4 0.06415190
epis.3      2      2      4 0.07837382
epis.4      2      1      2 0.07837382
$herit.cov
      fix.cov      ran.cov
Pheno.1  0.01603797  0.01796253
Pheno.2  0.01959345  0.02194467
$herit.gbye
      pheno qtl      add
GxE.1      1      1 0.05196304
GxE.2      1      3 0.05196304
GxE.3      2      3 0.06348279
GxE.4      2      6 0.06348279
$var
      Pheno.1  Pheno.2
Pheno.1      1.0      0.8
Pheno.2      0.8      1.0

```

The summary of the cross object can be viewed by:

```

> summary(cross)
Backcross
No. individuals:      200
No. phenotypes:       4
Percent phenotyped:   98.5 96 96.5 98.5
No. chromosomes:      19
      Autosomes:      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19
Total markers:        209
No. markers:          11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11
Percent genotyped:    95
Genotypes (%):        AA:50.7  AB:49.3

```

3.2. Bayesian QTL Analysis

This section describes in greater detail how to create Markov chain Monte Carlo (MCMC) samples from the Bayesian posterior to be used for QTL mapping. The function `qb.genoprob` creates a grid of putative QTL positions throughout the entire genome and computes the genotypic probabilities of the same. It would also compute genotypic probabilities for missing markers. By specifying a positive integer value for the option `step`, a grid of putative QTL can be created across the entire genome at intervals of the size of `step`. The genotypes at these putative loci are unobserved but the probabilities can be calculated from the observed marker data using the multipoint method (Jiang and Zeng).

```
> cross <- qb.genoprob(cross, map.func="haldane", step=1)
```

3.2.1. Specifying the Data—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 for non-normal continuous traits. The traits to be analyzed and the fixed and random covariates are specified by providing the corresponding element of `cross$pheno`. To jointly analyze multiple traits the option `multiple.trait` must be turned on. The option `multiple.type` specifies the type of multivariate model that is to be used. There are two choices, the Seemingly Unrelated Regression model (SUR) or the Traditional Multivariate model (TMV). SUR permits each trait to have a different genetic model while TMV assumes the same model for all traits (see (Banerjee *et al.*) for details). Note that, the cross object can have several phenotypes some of which could be used as covariates.

```
> data <- qb.data(cross, pheno.col = c(3,4), trait="normal",
multiple.trait=TRUE, fixcov=1, rancov=2, multiple.type="SUR", standardize =
FALSE)
```

3.2.2. Defining the model—The function `qb.model` defines the model for the QTL analysis. It specifies (i) if epistasis is considered, (ii) the *a priori* expected number of main effect QTL (`main.qtl`) for each trait, (iii) the *a priori* expected number of total number of QTLs for each trait included in the model through main, epistatic and gene-environment (G×E) interaction effect (`mean.qtl`) and (iv) the maximum number of QTLs allowed per chromosome (`chr.nqtl`). The maximum number of QTLs allowed (`max.qtl`) has a default of $l_0 + 3\sqrt{l_0}$ where l_0 is `main.qtl` in a non-epistatic and `mean.qtl` in an epistatic model. The interval (`interval`) between two flanking QTLs for each chromosome and the fixed covariate(s) interacting with QTLs are also specified. The `main.qtl` option of the `qb.model` function refers to the prior number of main effect QTL. This is obtained generally from a previous rudimentary non-Bayesian analysis for each trait and the maximum among the traits can be considered. For example, `R/qtl` can be used to analyze this data without epistasis and the number of QTLs detected could be used as the prior number of QTLs. The option `intcov` can be used to specify which fixed covariates are considered

for G×E interaction. If `multiple.type= "SUR"` in `qb.data`, the option `diff.loc` is specified to indicate if different QTL locations are considered for different traits (`diff.loc=TRUE`) or same set of locations are considered for each trait (`diff.loc=FALSE`). For more details please refer to (Banerjee *et al.*). The following chunk of code specifies the model for our simulated data:

```
> model <- qb.model(cross, epistasis = TRUE, intcov = 1, main.qtl = 3,
+                               mean.qtl = 6, interval =
rep(5,nchr(cross)),
+                               chr.nqtl = rep(2,nchr(cross)), max.qtl =
NULL,
+                               diff.loc = TRUE)
```

3.2.3. Generating MCMC samples—The function `qb.mcmc` creates MCMC samples on the data and model specified. The results are saved in a specified directory, default being the current working directory (“.”). The `genoupdate` parameter of the `qb.mcmc` function should be switched on when there are many missing marker genotypes. When this option is switched off a Haley-Knott regression is performed with the expected value of genotypes. For our case, we have only 7% of the marker genotypes missing at random, hence updating genotypes is not imperative. We would be saving 3000 samples which is the default `n.iter`. It is worth mentioning that `n.iter` does not specify the total number samples drawn from the posterior distribution but merely indicates the number of samples that would be saved and used for posterior analysis. The first `n.burnin` samples are discarded to allow the chain to converge to the posterior distribution and after that every `n.thin` sample is saved to reduce serial correlation among consecutive draws resulting in a total of `n.burnin + n.iter*n.thin` samples drawn.

```
> qbObject <- qb.mcmc(cross, data, model, n.burnin = 5000, n.thin = 20,
+                               n.iter = 3000, genoupdate=FALSE)
qb.mcmc is running 65,000 iterations. The current iterations are saved:
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
```

```

2800
3000
MCMC sample has been saved to: ./pheno.normal.1_Jul-03-223638.
Bayesian MCMC took 6.18 minutes.

```

3.2.4. Examining the qb Object—Please note that the plot and summary routines in this package all use the S3 generic method. That is, we create an object with a call to `qb.xxx` and then plot it using the generic `plot` command. The `qbObject` is an object of class `qb` to which we can apply the generic `summary` or `plot` routines. We defer plots to later sections. The summary of MCMC posterior samples can be summarized by trait.

```

> summary(qbObject,pheno.col=1)
Bayesian model selection QTL mapping object qbObject on cross object cross
had 3000 iterations recorded at each 20 steps
with 5000 burn-in steps.
MCMC runs saved in qb object.
Traits pheno.normal.1,pheno.normal.2 ( 1,2 ) treated as normal.
Fixed covariate: fix.cov
Random covariate: random.cov
Trait was not standardized.
Epistasis was allowed.
Prior number of QTL: 3 main, 6 total, with 13 maximum.
Minimum distance between QTL:
      [1] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
Maximum number of QTL:
      [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
QTL by environment allowed.
Interacting covariates: 1
Diagnostic summaries:

```

		nqtl	mean	envvar	varadd	varaa	envadd
varenv	var						
Min.	1.000	8.032	0.9605	0.2748	0.1655	0.01705	-1.8930 0.832
1st Qu.	4.000	9.272	1.3490	0.5303	0.3626	0.22810	0.6967 1.253
Median	5.000	9.571	1.4570	0.6038	0.4230	0.31520	0.8843 1.373
Mean	5.419	9.572	1.4690	0.6110	0.4312	0.36920	0.8500 1.411
3rd Qu.	6.000	9.856	1.5750	0.6821	0.4952	0.43910	1.0440 1.515
Max.	11.000	11.510	2.3490	1.2610	0.8201	2.87200	1.5620 4.112

```

Percentages for number of QTL detected:
1  2  3  4  5  6  7  8  9  10  11
0  2  8 20 25 22 14  6  2  0  0
Percentages for number of epistatic pairs detected:
pairs
2  3  4  5  6  7
4 28 36 21  8  2

```

Percentages for common epistatic pairs:

1:3	1:2	1:7	1:1	1:14	3:9	2:3	2:7	2:12	12:14	1:8
7:14	102									
101	90	35	5	4	4	3	2	2		
2	2									
2:11	4:7	3:13								
2	2	1								

3.2.5. Plotting MCMC History—Convergence of the MCMC chain is of considerable interest to any Bayesian analysis. The plotting tools in `R/qtlbim` provide a method for visually inspecting the history of the MCMC run. The command

```
plot(qb.coda(qbObject,phenol.col=1))
```

shows a default view of the MCMC chain as a time series for a particular trait. Since each iteration of the MCMC chain represents a single model for a particular trait; 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 (`nqtl`) and the model means, use the following command. The results of the following command are shown in Figure 1.

```
> plot(qb.coda(qbObject,variables=c("nqtl","envvar"),phenol.col=1))
```

3.2.6. Plot of QTL 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 for a particular trait. The QTL shown 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(qbObject,pheno.col=1))
```

produces a plot shown in Figure 2 for the first trait.

3.2.7. Summary Plot for Sampled Models—The function `qb.BayesFactor` produces a composite (4-by-2) summary plot of the models sampled by the MCMC chain for a particular trait. 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 3 shows the plot produced by the command `qb.BayesFactor(qbObject, phenol.col=1)`. 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. The third row addresses the frequency of sampling each chromosome. The fourth row shows relative importance of epistatic pairs. Here the “1:3”, or chr 1 by chr 3, epistatic pair is by far the strongest.

3.3. One Dimensional (1-D) and Two-dimensional (2-D) Genome Scans

In the present context, the term “scan” refers to methods based on constructing one or two dimensional profiles of QTL likelihoods or posterior distributions.

3.3.1. 1-D Scan—For every locus in the genome, it is meaningful to examine the chances of this position being a true QTL. Being more specific, to test the j^{th} position for each trait t , we can compute the Bayes Factor (BF) as:

$$BF = \frac{p(\gamma_{tj}=1|y_t)/p(\gamma_{yj}=1)}{p(\gamma_{tj}=0|y_t)/p(\gamma_{yj}=0)}$$

In practice these marginal posterior probabilities (*e.g.* $p(\gamma_{tj}=1|y_t)$) are computed using the idea of model averaging. $p(\gamma_{tj}=1|y_t)$ is the proportion of sampled models which have included the current position of interest. In essence, we not only consider the one-QTL models with $\gamma_{tj}=1$ only, but also any q -QTL model which includes the current position and hence we average over different models. Although it might seem paradoxical to compute BF for 1 vs 0 QTL in a multiple QTL mapping framework, by virtue of model averaging we can actually control for QTL elsewhere in the genome. This idea can be extended to investigating contribution in terms of heritability or parameter effects. However, there are some mechanics involved, for instance, scans involving epistatic effects need to consider each pair for both loci. This is termed as a two-dimensional scan.

3.3.2. 2-D Scan—A two-dimensional scan involves considering models including a pair of loci (j_1, j_2) to models including either or none. It is important to recall that a locus can be included in the model through its marginal (additive, dominant) and/or epistatic (additive-additive, additive-dominant, dominant-additive, dominant-dominant) effects. BF can be examined for any combination of these effects. One useful hypothesis could be testing the “full” model (marginal + epistatic effects) *vs* epistatic effects alone.

3.3.3. Using `qb.scan` routines for multiple traits—There are two scan routines in R/ qtlbim corresponding to a 1-D or 2-D scan, namely, `qb.scanone` and `qb.scantwo`. They are collectively termed as “qb.scan” routines. Each iteration of the Monte Carlo chain contained in `qbObject` 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 posterior profile of a particular trait, or the posterior probability of a QTL at each locus for the trait.

```
> one <- qb.scanone(qbObject, type="posterior", pheno.col=1)
> one
```

	posterior	of pheno.normal.1		for main,GxE,epistasis,sum			
		n.qtl	pos	m.pos	e.pos	main	GxE
epistasis							
			sum				
1	2.0000	17	17	17	0.299880	0.299982	0.29934
	0.29998						
2	1.0675	11	11	11	0.112932	0.111697	0.16639
	0.16643						
3	1.0745	13	14	13	0.008585	0.010559	0.11452
	0.11461						
4	0.1186	72	72	74	0.004921	0.000538	0.00152
	0.00543						
5	0.1083	87	56	87	0.000909	0.000492	0.00257
	0.00259						
6	0.2042	82	82	94	0.008741	0.001142	0.00293
	0.00943						
7	0.1003	97	5	97	0.001336	0.000417	0.00303
	0.00315						
8	0.0759	24	71	85	0.001021	0.000456	0.00200
	0.00181						
9	0.0712	0	92	29	0.000809	0.000668	0.00150
	0.00200						
10	0.0695	3	10	2	0.001168	0.000985	0.00167
	0.00219						
11	0.1758	49	65	49	0.001006	0.000334	0.00686
	0.00692						
12	0.0685	33	30	53	0.000958	0.000340	0.00200
	0.00197						

13	0.2727	45	45	41	0.008578	0.000753	0.00204
0.00898							
14	0.0755	46	15	46	0.001172	0.000521	0.00209
0.00234							
15	0.1197	50	80	50	0.000915	0.000553	0.00497
0.00519							
16	0.0849	87	25	87	0.001266	0.000365	0.00182
0.00201							
17	0.2253	14	15	26	0.004564	0.001503	0.00225
0.00562							
18	0.0816	8	0	18	0.000840	0.000419	0.00153
0.00170							
19	0.0702	7	23	81	0.000970	0.000334	0.00167
0.00194							

Here we show $2\log(\text{BF})$, or 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, 2, 3 and 4 more closely, we can plot subsets of chromosomes by using the plot command (Figure 4).

```
> one <- qb.scanone (qbObject,type="2logBF",pheno.col=1)
> plot (one,chr=1:4)
```

As shown in Figure 4, the $2\log(\text{BF})$ profile of the first trait can be obtained for all types of effect, namely, main effect, epistatic and GxE interactions. It might be of interest to look at main effects of all traits simultaneously in a single graphical device. To do that one can run the following chunk of code and get a plot shown in Figure 5.

```
>plot(qb.scanone(qbObject,type="heritability",pheno.col=2),scan="main",
+      chr=1:4,main="heritability of all traits for main")
>plot(qb.scanone(qbObject,type="heritability",pheno.col=1),scan="main",
+      chr=1:4,add=TRUE,col="red")
```

The function `qb.scantwo` gives a two dimensional scan that allows us to look for possible epistatic effects between putative QTL. We can use `qb.scantwo` to examine the $2\log(\text{BF})$ of pair of QTL for a particular trait (the first trait is shown here) using the following chunk of code.

```
> two <- qb.scantwo(qbObject,type="2logBF",pheno.col=1,verbose=T)
finding all pairs of loci ...
upper: 2logBF of pheno.normal.1 for aa
lower: 2logBF of pheno.normal.1 for add+aa+add.fix.cov
non-epistatic components ...
```

```

epistatic    components ...
qb.scanone on diagonal with add ...
> plot(two,chr=1:4)

```

Figure 6 shows the plot of `qb.scantwo` results. The upper triangle indicates the epistatic effects between pairs of loci for a particular trait and the lower triangle indicates the full model, i.e, main + epistatic + G×E for the corresponding pair influencing a particular trait. As can be seen in Figure 6, the upper triangle shows two hot spots indicating epistasis between chromosome pair 1×2 and 1×3.

3.3.4. Types of Scan Summaries—There are several types of scan summaries, illustrated below. These include the following LPD, heritability, variance components, parameter estimates, cell means, posterior probabilities and Bayes factors. For each type, we can provide a summary scan, and in addition provide detail broken down by main effects, epistatic effects, and/or G×E (genotype by environment, or genotype by covariate) interactions. These breakdowns can be further divided into Cockerham (Cockerham ; Kao and Zeng) type effects (additive and dominance for main effects, or the four epistatic interactions of additive-additive (aa), additive-dominant (ad), dominant-additive (da), and dominant-dominant (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 G×E 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 explained variation.
- `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). The LPD for QTLs was introduced by (Sen and Churchill). It tests presence or absence of a QTL at a locus, adjusting for all other possible model effects (other QTL, epistasis and G×E).
- `detection` is the posterior probability of detection of a QTL at a

3.4. Testing Pleiotropy

It is of considerable biological interest to test if a QTL co-maps to more than one trait. This is known as pleiotropy. The function `qb.conjoin` finds the posterior probability (or Bayes Factors) that multiple traits co-map to the same locus. The function `qb.conjoin` as default calculates the posterior probability that a particular locus maps to any 1, any 2, ... all traits. This can be controlled by using the option `set.size`. The following chunk of code illustrates this:

```
> con <- qb.conjoin(qbObject)
> plot(con,size=1)
```

Figure 7 shows the posterior probability that a QTL maps to exactly one trait (indicated by the option `size` in the plot command). This is an alternate way of initial examination of the MCMC samples. Figure 8 shows the posterior probability that a QTL maps to exactly two traits and hence a pleiotropic QTL. The can be done by using the same `conjoin` object and using the following command.

```
> plot(con,size=2)
```

Figure 8 indicates that chromosome 2 has the strongest pleiotropic QTL. It is worth mentioning that pleiotropic events are rare and it is recommended to run a longer chain (`n.iter = 10000`) to test pleiotropic effects.

Acknowledgments

We would like to thank Dr. Brian Yandell, University of Wisconsin-Madison, for his immense help in the development of the software, especially the visualization of posterior samples and developing the `qb.conjoin` routine. SB's work was supported by the NIH Clinical Translational Science Center grant UL1-RR024996, and NY's work was supported by NIH R01 GM069430.

References

1. Banerjee S, Yandell BS, Yi N. Bayesian quantitative trait loci mapping for multiple traits. *Genetics*. 2008; 179:2275–2289. [PubMed: 18689903]
2. Banerjee S, Yandell BS, Yi N. 2010 Bayesian Genome-Wide Interacting QTL Mapping for Multiple Traits. In Review
3. Broman KW, Speed TP. A model selection approach for the identification of quantitative trait loci in experimental crosses. *Journal of Royal Statistical Society B*. 2002; 64:641–656.
4. Broman KW, Wu H, Sen S, Churchill GA. R/qtl: QTL mapping in experimental crosses. *Bioinformatics*. 2003; 19:889–890. [PubMed: 12724300]
5. Cockerham CC. An Extension of the Concept of Partitioning Hereditary Variance for Analysis of Covariances among Relatives When Epistasis Is Present. *Genetics*. 1954; 39:859–882. [PubMed: 17247525]
6. Jiang C, Zeng ZB. Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics*. 1995; 140:1111–1127. [PubMed: 7672582]
7. Jiang C, Zeng ZB. Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. *Genetica*. 1997; 101:47–58. [PubMed: 9465409]

8. Kao CH, Zeng ZB. Modeling epistasis of quantitative trait loci using Cockerham's model. *Genetics*. 2002; 160:1243–1261. [PubMed: 11901137]
9. Lander ES, Botstein D. Mapping mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics*. 1989; 121:185–199. [PubMed: 2563713]
10. Plummer M, Best N, Cowles K, Vines K. CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News*. 2006; 6:7–11.
11. Sen S, Churchill GA. A statistical framework for quantitative trait mapping. *Genetics*. 2001; 159:371–387. [PubMed: 11560912]
12. Yandell BS, Mehta T, Banerjee S, Shriner D, Venkataraman R, et al. R/qtlbim: QTL with Bayesian Interval Mapping in experimental crosses. *Bioinformatics*. 2007; 23:641–643. [PubMed: 17237038]
13. Yi N. A unified Markov chain Monte Carlo framework for mapping multiple quantitative trait loci. *Genetics*. 2004; 167:967–975. [PubMed: 15238545]
14. Yi N, Yandell BS, Churchill GA, Allison DB, Eisen EJ, et al. Bayesian model selection for genome-wide epistatic quantitative trait loci analysis. *Genetics*. 2005; 170:1333–1344. [PubMed: 15911579]

Figure 1.
Diagnostic plot for a MCMC run.

Figure 2. A jittered plot of QTL included in the model for the first trait through main, epistatic or G×E interaction.

Figure 3.

Left Panels: Paired plots of posteriors (as bars) overlaid by priors (as blue lines). Right panels: Bayes factor ratios to the least likely model. 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.

Figure 4.

The `qb.scanone` results of the first trait for all effects restricted to chromosomes 1, 2, 3 and 4.

Figure 5.

The `qb.scanone` results of all traits for main effects restricted to chromosomes 1, 2, 3 and 4. Red lines indicate “heritability” profile for the first trait and black lines indicate the same for the second trait.

Figure 6. Plot of `qb.scantwo` results for the first trait. Upper triangle indicates $2\log(\text{BF})$ for epistatic effects between pairs of loci and lower triangle indicates the same for main + epistatic + $G \times E$ interaction effects between the pairs of loci.

Figure 7.

Posterior profile that a QTL map to exactly one trait. Black lines indicate posterior profile for the first trait and red lines indicate the same for the second trait.

Figure 8.
Posterior profile that a QTL maps to both traits indicating pleiotropy.