

Single-QTL analysis

The most commonly used method for QTL analysis is interval mapping, in which one posits the presence of a single QTL and considers each point on a dense grid across the genome, one at a time, as the location of the putative QTL. A central issue concerns the treatment of missing genotype information: at a position between genetic markers, genotype data are not available and must be inferred on the basis of the available marker genotype data. Several methods are available; we describe the most popular. These methods all have analogs for the fit of multiple-QTL models, which will be discussed in Chap. 8 and 9. We further discuss the establishment of statistical significance in such single-QTL genome scans, and the special treatment that is required for the X chromosome. But first, in order to introduce the basic ideas in QTL mapping, we describe an even simpler method, sometimes called marker regression.

Each section will begin with a bit of theory, followed by R code for performing the analyses with R/qtl. The R code is cumulative through the chapter; the results in one section may rely on code executed in a previous section.

4.1 Marker regression

The simplest method for the analysis of QTL mapping data is to consider each marker individually, split the individuals into groups, according to their genotypes at the marker, and compare the groups' phenotype averages. While this method can seldom be recommended for use in practice, it provides a valuable framework for thinking about QTL mapping and for describing some of the essential issues in QTL mapping.

Consider, for example, the **hyper** data, described in Sec. 2.3. The blood pressure phenotype is plotted against the genotype at markers D4Mit214 and D12Mit20 in Fig. 4.1. At D4Mit214, the homozygous individuals exhibit a larger average phenotype than the heterozygotes, indicating that this marker is linked to a QTL. At D12Mit20, on the other hand, the two genotype groups

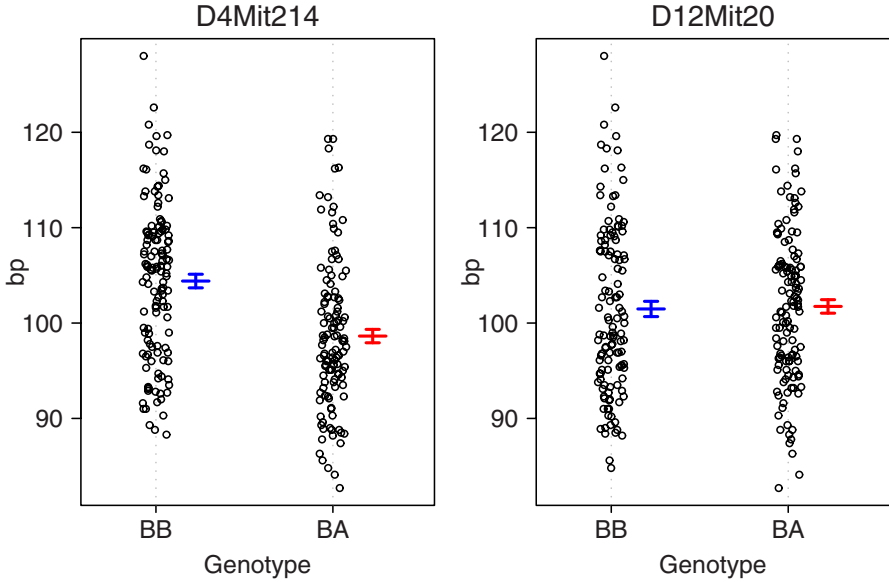


Figure 4.1. Dot plots of the blood pressure phenotype against the genotype at two selected markers, for the **hyper** data. Confidence intervals for the average phenotype in each genotype group are shown.

show similar phenotypes, and so D12Mit20 is not indicated to be linked to a QTL.

In a backcross, we test for linkage of a marker to a QTL by a t test; in an intercross, we would use analysis of variance (ANOVA), which gives an F statistic. Traditionally, evidence for linkage to a QTL is measured by a LOD score: the $\log_{10}$ likelihood ratio comparing the hypothesis that there is a QTL at the marker to the hypothesis that there is no QTL anywhere in the genome.

The LOD score at a marker is calculated as follows. First, consider the null hypothesis of no QTL, in which case, with y_i denoting the phenotype for individual i , $y_i \sim N(\mu, \sigma^2)$ (i.e., the phenotypes follow a single normal distribution, independent of the genotypes). We consider the likelihood function $L_0(\mu, \sigma^2) = \Pr(\text{data} \mid \text{no QTL}, \mu, \sigma^2) = \prod_i \phi(y_i; \mu, \sigma^2)$, where ϕ is the density of the normal distribution. We take as estimates of μ and σ^2 the values for which the likelihood is maximized (such estimates are called the maximum likelihood estimates, MLEs). For this model, the MLE of μ is simply the phenotype average, $\bar{y}$, and the MLE of σ^2 is RSS_0/n , where $\text{RSS}_0 = \sum_i (y_i - \bar{y})^2$ is the null residual sum of squares and n is the sample size. The $\log_{10}$ likelihood for the null hypothesis is obtained by plugging in the MLEs; with a bit of algebra, it reduces to $-\frac{n}{2} \log_{10} \text{RSS}_0$.

Under the alternative hypothesis, that there is a QTL at the marker under test, we assume that $y_i | g_i \sim N(\mu_{g_i}, \sigma^2)$, where g_i is the genotype of

individual i at the marker, μ_{AA} and μ_{AB} are the phenotype averages for the two genotype groups, and σ^2 is the residual variance (assumed to be the same in the two groups). The likelihood function is $L_1(\mu_{AA}, \mu_{AB}, \sigma^2) = \Pr(\text{data} | \text{QTL at marker}, \mu_{AA}, \mu_{AB}, \sigma^2) = \prod_i \phi(y_i; \mu_{g_i}, \sigma^2)$. We again estimate the parameters by maximum likelihood: the values for which L_1 achieves its maximum. The MLEs for the μ_i are simply the phenotype averages within the two genotype groups. The MLE of σ^2 is the pooled estimate, RSS_1/n , where $\text{RSS}_1 = \sum_i (y_i - \hat{\mu}_{g_i})^2$ is the residual sum of squares under the alternative. The $\log_{10}$ likelihood for the alternative hypothesis is obtained by plugging in the MLEs; it reduces to $-\frac{n}{2} \log_{10} \text{RSS}_1$.

Finally, the LOD score is the difference between the $\log_{10}$ likelihood under the alternative hypothesis and the $\log_{10}$ likelihood under the null hypothesis, and so we obtain the following.

$$\text{LOD} = \frac{n}{2} \log_{10} \left(\frac{\text{RSS}_0}{\text{RSS}_1} \right)$$

Note that individuals with missing genotype data at the marker must be omitted from the estimation under the alternative, and so, in the calculation of the LOD score, such individuals are omitted from both the alternative and null likelihoods.

The LOD score is equivalent to the F statistic from ANOVA. (And note that the t statistic in a backcross is the signed square root of the F statistic that would be obtained from ANOVA.) Let df denote the degrees of freedom ($\text{df}=1$ for a backcross and $\text{df}=2$ for an intercross); the connection between the F statistic and the LOD score is as follows.

$$\begin{aligned} F &= \left(\frac{\text{RSS}_0 - \text{RSS}_1}{\text{RSS}_1} \right) \left(\frac{n - \text{df} - 1}{\text{df}} \right) \\ &= \left(\frac{\text{RSS}_0}{\text{RSS}_1} - 1 \right) \left(\frac{n - \text{df} - 1}{\text{df}} \right) \\ &= \left(10^{\frac{2}{n} \text{LOD}} - 1 \right) \left(\frac{n - \text{df} - 1}{\text{df}} \right) \end{aligned}$$

The inverse of this formula is also of interest.

$$\text{LOD} = \frac{n}{2} \log_{10} \left[F \left(\frac{\text{df}}{n - \text{df} - 1} \right) + 1 \right]$$

Note further that the estimated proportion of the phenotypic variance explained by the QTL (i.e., the estimated heritability due to the QTL) is $(\text{RSS}_0 - \text{RSS}_1)/\text{RSS}_0$. Thus, the estimated percent variance explained by the QTL is $1 - 10^{-\frac{2}{n} \text{LOD}}$.

Large LOD scores indicate evidence for the presence of a QTL, but in considering the statistical significance of LOD scores, we must take account of the multiple tests performed. Discussion of this issue is deferred to Sec. 4.3.

The key advantage of marker regression is its simplicity: we just perform a t test or ANOVA at each marker. Thus, no special software is required, and one may easily incorporate covariates (such as sex) or extend the analysis to more complex models (such as for the treatment of censored survival times).

A key disadvantage is that one must omit individuals with missing marker genotypes. Further, one cannot inspect positions between markers, and one obtains rather poor information about QTL location. Also, the apparent effect of a QTL is attenuated by its incomplete linkage to a marker. For example, consider a backcross with a single QTL, and let μ_{AA} and μ_{AB} denote the phenotype averages for the two QTL genotypes, so that the effect of the QTL is $\Delta = \mu_{AB} - \mu_{AA}$. If the recombination fraction between a marker and the QTL is r , then the individuals with marker genotype AA will consist of a fraction $(1 - r)$ with QTL genotype AA and a fraction r with QTL genotype AB. Thus the average phenotype for individuals that are AA at the marker will be $\mu_{AA}(1 - r) + \mu_{AB}r$. Similarly, the average phenotype for individuals that are AB at the marker will be $\mu_{AB}(1 - r) + \mu_{AA}r$. As a result, the difference between the phenotype averages for the two marker genotype groups will be $[\mu_{AB}(1 - r) + \mu_{AA}r] - [\mu_{AA}(1 - r) + \mu_{AB}r] = \Delta(1 - 2r)$, and so the apparent effect of the QTL is reduced by a factor $(1 - 2r)$ due to its incomplete linkage to the marker.

The most important disadvantage of marker regression is that we consider the presence of a single QTL. Thus we have limited ability to separate linked QTL and no ability to assess possible interactions among QTL. This disadvantage is shared with all of the methods described in this chapter.

Example

Let us now turn to the actual analysis with R/qtl, using the `hyper` data as an example. We first load R/qtl and the data.

```
> library(qtl)
> data(hyper)
```

First note that the dot plots of phenotype against genotype, in Fig. 4.1, were created with the `plot.pxd` function, as follows.

```
> par(mfrow=c(1,2))
> plot.pxd(hyper, "D4Mit214")
> plot.pxd(hyper, "D12Mit20")
```

The fit of single-QTL models is accomplished with the function `scanone`. The use of the argument `method="mr"` indicates to use marker regression (i.e., ANOVA or a t test at each marker). For the `hyper` data, we type the following.

```
> out.mr <- scanone(hyper, method="mr")
```

The result, saved in `out.mr`, is a matrix with three columns: chromosome, cM position, and LOD score. We can look at the results for the chromosome 12 as follows.

```
> out.mr[ out.mr$chr == 12, ]
```

	chr	pos	lod
D12Mit37	12	1.1	0.3610905
D12Mit110	12	16.4	0.0009559
D12Mit34	12	23.0	0.0005335
D12Mit118	12	40.4	0.0003868
D12Mit20	12	56.8	0.0136116

The output of `scanone` has class `"scanone"`, and so use of the functions `plot` and `summary` with the `out.mr` object will create a plot or summary via `plot.scanone` and `summary.scanone`, respectively. For example, we may pull out the single largest LOD score from each chromosome with `summary.scanone`; we show just those having LOD > 3 with the `threshold` argument.

```
> summary(out.mr, threshold=3)
```

	chr	pos	lod
D1Mit14	1	82.0	3.52
D4Mit214	4	21.9	6.86

The function `max.scanone` may be used to pick out the single biggest LOD score, as follows.

```
> max(out.mr)
```

	chr	pos	lod
D4Mit214	4	21.9	6.86

A plot of the LOD scores for chromosomes 4 and 12 is obtained with the following. The result appears in Fig. 4.2.

```
> plot(out.mr, chr=c(4, 12), ylab="LOD score")
```

The argument `chr` is used to select the chromosomes to plot; by default all chromosomes will be plotted. The argument `ylab` is used to change the *y*-axis label.

The jagged appearance of the LOD curve for chromosome 4 is due to the pattern of missing marker genotype data. Recall that in the marker regression method, we split the individuals into groups according to their genotype at a marker. If an individual's genotype is missing, that individual must be omitted, as we do not know the genotype group in which it should be placed. At some markers in the `hyper` data, only recombinant individuals were genotyped (see Fig. 1.7 on page 10), and so these markers, in isolation, will provide little evidence for linkage to a QTL.

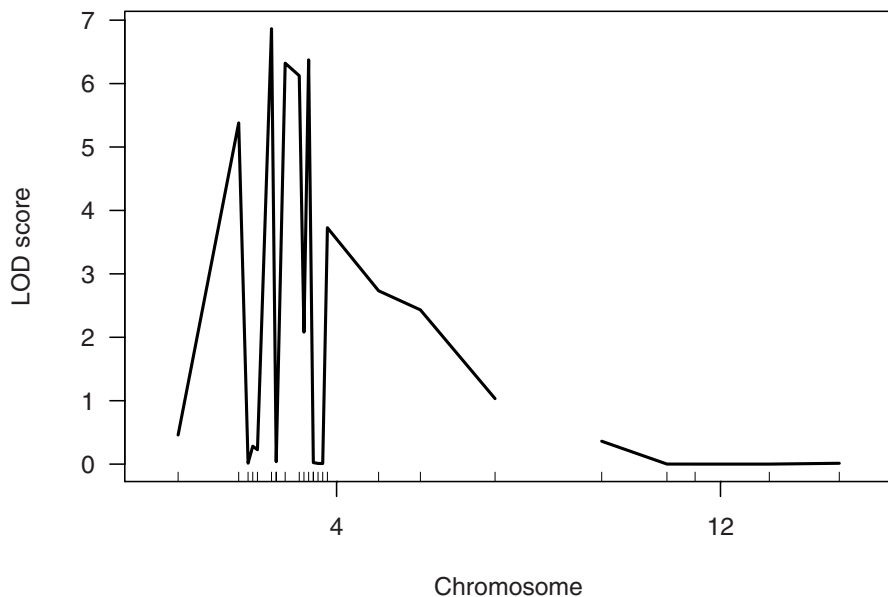


Figure 4.2. LOD scores for each marker on chromosomes 4 and 12 for the **hyper** data, calculated by marker regression.

4.2 Interval mapping

In this section, we consider several variants on interval mapping. Interval mapping improves on the marker regression method by taking account of missing genotype data at a putative QTL. The various interval mapping methods differ in their treatment of the missing genotype data. Standard interval mapping uses maximum likelihood estimation under a mixture model, while the Haley–Knott regression methods use approximations to the mixture model. The multiple imputation method uses the same mixture model but with multiple imputation in place of maximum likelihood.

4.2.1 Standard interval mapping

In standard interval mapping, we again assume the presence of a single QTL, but now we consider a grid of positions along the genome as the possible locations for the QTL. Consider a particular fixed position for the QTL, and let g_i denote the QTL genotype for individual i . As with the marker regression method, we assume that $y_i|g_i \sim N(\mu_{g_i}, \sigma^2)$, but now the QTL genotype is generally not known.

For each individual, we may calculate $p_{ij} = \Pr(g_i = j|\mathbf{M}_i)$, where $\mathbf{M}_i$ denotes the multipoint marker genotype data for individual i . For example, consider a backcross and two markers separated by a recombination fraction

Table 4.1. Conditional probabilities for the QTL genotypes in a backcross, given the genotypes at two flanking markers.

Marker genotype		QTL genotype	
Left	Right	AA	AB
AA	AA	$(1 - r_{1Q})(1 - r_{2Q})/(1 - r_{12})$	$r_{1Q}r_{2Q}/(1 - r_{12})$
AA	AB	$(1 - r_{1Q})r_{2Q}/r_{12}$	$r_{1Q}(1 - r_{2Q})/r_{12}$
AB	AA	$r_{1Q}(1 - r_{2Q})/r_{12}$	$(1 - r_{1Q})r_{2Q}/r_{12}$
AB	AB	$r_{1Q}r_{2Q}/(1 - r_{12})$	$(1 - r_{1Q})(1 - r_{2Q})/(1 - r_{12})$

of r_{12} . Suppose that our putative QTL sits in the intervening interval, and let r_{iQ} denote the recombination fraction between marker i and the QTL. If we assume no crossover interference and no genotyping errors, and if an individual is typed at both markers, the QTL genotype probabilities given the marker genotypes are as in Table 4.1. In general, one may use algorithms for hidden Markov models (HMMs) for these sorts of calculations, as one can then allow for the presence of genotyping errors and more simply deal with partially informative genotypes (such as the case of dominant markers in an intercross). For further details, see Sec. 1.3.1 and Appendix D.

Given the marker data, an individual's phenotype follows a *mixture* of normal distributions, with known mixing proportions (the p_{ij}). That is, the density function for the phenotype of individual i is $\sum_j p_{ij} \phi(y_i; \mu_j, \sigma^2)$, where ϕ is the density of the normal distribution and the sum is over the possible QTL genotypes.

For example, consider a backcross containing a single QTL, and consider two markers separated by 20 cM, with the QTL placed in the intervening interval, 7 cM from the left marker. The phenotype distributions, conditional on the genotypes at the two markers, are shown in Fig. 4.3.

Except in the case of rare double recombination events, all individuals who are AA at both markers (top panel) will also be AA at the QTL, and so their phenotypes will follow a normal distribution centered at μ_{AA} . Similarly, individuals who are AB at both markers (lower panel) will generally also be AB at the QTL, and so their phenotypes will follow a normal distribution centered at μ_{AB} .

Individuals who are AA at the left marker but AB at the right marker will consist of some individuals who are AA at the QTL (i.e., their recombination event occurred to the right of the QTL) and some who are AB at the QTL (having recombined to the left of the QTL). Similarly, individuals who are AB at the left marker but AA at the right marker will consist of some individuals who are AB at the QTL (having recombined to the right of the QTL) and some who are AA at the QTL (having recombined to the left of the QTL). The dashed curves in the central panels in Fig. 4.3 are the distributions of the groups with common QTL genotype; the solid curves are the mixture distributions.

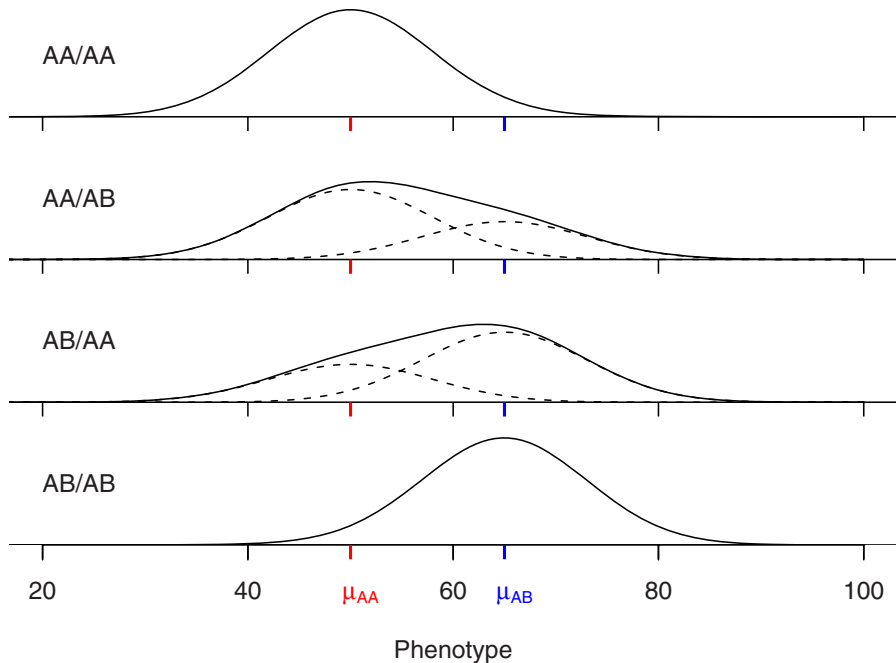


Figure 4.3. Phenotype distributions conditional on the genotype at two markers, in the case of a backcross containing a single QTL. The markers are separated by 20 cM and the QTL sits within the marker interval, 7 cM from the left marker. The dashed curves are the distributions of the groups with common QTL genotype; the solid curves are the mixture distributions.

We estimate the μ_j and σ by maximum likelihood; that is, we take as our estimates those values for which the observed data is most probable. The likelihood function is $L(\boldsymbol{\mu}, \sigma) = \prod_i \sum_j p_{ij} \phi(y_i; \mu_j, \sigma^2)$, where the sum is over the possible QTL genotypes. The MLEs cannot be obtained in closed form; an iterative algorithm is necessary. We use a form of the EM algorithm. We begin with initial estimates $\hat{\mu}_j^0$ and $\hat{\sigma}^{(0)}$.

In the E-step at iteration s , we calculate the conditional probability that an individual is in QTL genotype group j given its marker data, phenotype, and our current estimates of the μ_j and σ .

$$\begin{aligned} w_{ij}^{(s)} &= \Pr(g_i = j | \mathbf{M}_i, y_i, \hat{\mu}_j^{(s-1)}, \hat{\sigma}^{(s-1)}) \\ &= \frac{p_{ij} \phi(y_i; \hat{\mu}_j^{(s-1)}, \hat{\sigma}^{(s-1)})}{\sum_k p_{ik} \phi(y_i; \hat{\mu}_k^{(s-1)}, \hat{\sigma}^{(s-1)})} \end{aligned}$$

In the M-step, we update our estimates of the μ_j and σ , treating the $w_{ij}^{(s)}$ as weights.

$$\hat{\mu}_j^{(s)} = \sum_i w_{ij}^{(s)} y_i / \sum_i w_{ij}^{(s)}$$

$$\hat{\sigma}^{(s)} = \sqrt{\frac{\sum_{ij} w_{ij}^{(s)} (y_i - \hat{\mu}_j^{(s)})^2}{n}}$$

Iterations are repeated until the estimates converge (i.e., until the estimates stop changing). The EM algorithm has the advantage that the likelihood is nondecreasing across iterations. It may be that the algorithm converges to a local maximum, but with relatively dense markers and relatively complete marker genotype data, the likelihood is well behaved and the EM algorithm will converge to the global maximum. If there were multiple modes in the likelihood surface, one would want to use multiple initial estimates for the EM algorithm, but because the likelihood is well behaved in this context, we generally start the algorithm by taking $w_{ij}^{(0)} = p_{ij}$ and then doing the M-step. This is equivalent to using Haley–Knott regression (Sec. 4.2.2) to get the initial estimates.

Once the maximum likelihood estimates of the μ_j and σ have been obtained, a LOD score is calculated as follows.

$$\text{LOD} = \log_{10} \left(\frac{\prod_i \sum_j p_{ij} \phi(y_i; \hat{\mu}_j, \hat{\sigma}^2)}{\prod_i \phi(y_i; \hat{\mu}_0, \hat{\sigma}_0^2)} \right)$$

where $\hat{\mu}_0$ and $\hat{\sigma}_0$ are the average and SD of the y_i , so that the denominator of the LOD score is the likelihood under the null hypothesis that there is no QTL anywhere in the genome.

In standard interval mapping, the EM algorithm is performed at each position on a grid of putative QTL locations along the genome, while the estimates and likelihood under the null hypothesis are calculated just once.

The key advantage of interval mapping, relative to marker regression, is that one takes appropriate account of missing genotype information. Thus, we need not omit individuals with missing genotype at a marker, and we may inspect positions between markers. We thus obtain a more clear understanding of QTL location. (The smooth LOD curves are also nicer to look at. Before the development of interval mapping, papers reporting the results of QTL mapping contained long tables of p -values; the plot of LOD curves was an important advance.) Further, we may obtain an improved estimate of the effect of a QTL, as such an estimate may be obtained at its estimated position, rather than using the genotypes at the nearest genetic marker.

Disadvantages of interval mapping include increased computation time, the need for specialized software, and difficulty in generalizing the method for more complex models or for the inclusion of environmental and other covariates. These disadvantages are no longer of much importance, as interval mapping requires only a couple of seconds of computer time, a variety of relevant computer programs are available, and a variety of extensions of interval mapping have been implemented.

The most important disadvantage of interval mapping is that we are still considering only a single-QTL model, and so we have limited ability to separate linked QTL and no ability to assess possible interactions among QTL. With complete marker genotype data, interval mapping and marker regression give precisely the same results at the markers. Interval mapping seems fancy, but it is little different, conceptually, from simply performing ANOVA at each marker.

Example

Having completed our discussion of the theory underlying standard interval mapping, we now turn to the calculations in R/qtl. We first must calculate the conditional genotype probabilities, $p_{ij} = \Pr(g_i = j | \mathbf{M}_i)$. This is done with the function `calc.genoprob`. The argument `step` is used to define the density (in cM) of the grid on which these probabilities will be calculated; this will determine the density at which interval mapping is performed. The argument `error.prob` allows the probabilities to be calculated assuming a given rate of genotyping errors.

```
> hyper <- calc.genoprob(hyper, step=1, error.prob=0.001)
```

In `calc.genoprob`, one may also use the argument `off.end` to calculate the probabilities to some distance past the terminal markers on the chromosome, so that interval mapping will be performed past the terminal markers, but this can lead to the artifacts in the results (for example, a QTL near the end of the chromosome may show a mirror image past the terminal marker), and so we generally use `off.end=0` (the default).

We should emphasize that it is important, for analyses with R/qtl, that no two markers are placed at precisely the same position. If there are markers that coincide, a warning will be produced by the function `summary.cross`. Marker positions may be moved apart slightly with the function `jittermap`, as follows. Note that this should be done *prior* to the call to `calc.genoprob`.

```
> hyper <- jittermap(hyper)
```

Interval mapping is performed with the `scanone` function, using the argument `method="em"` (for “EM algorithm”).

```
> out.em <- scanone(hyper, method="em")
```

Standard interval mapping is the default method, and so `method="em"` can be omitted, as follows.

```
> out.em <- scanone(hyper)
```

The form of the results was described in the previous section. We can plot the results for all chromosomes as follows; the results appear in Fig. 4.4.

```
> plot(out.em, ylab="LOD score")
```

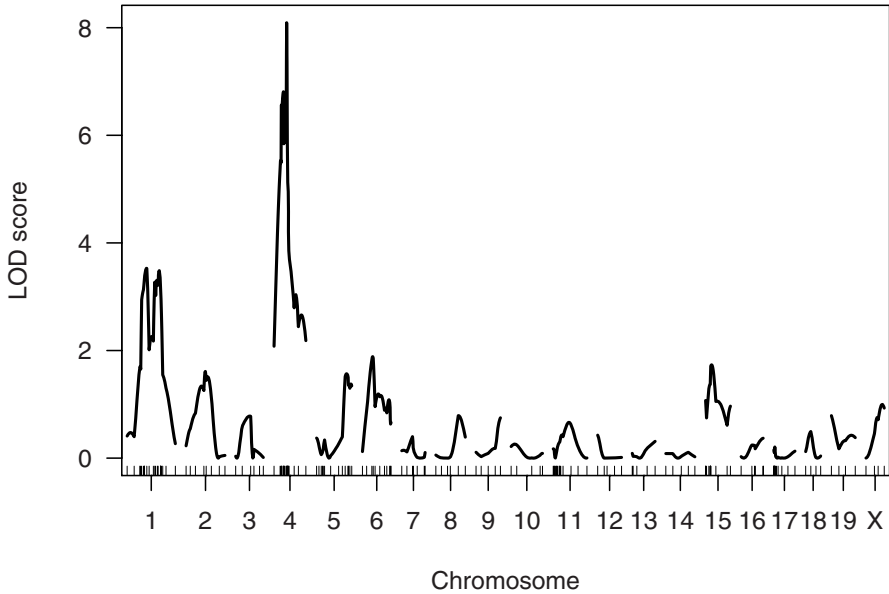


Figure 4.4. LOD scores by standard interval mapping for the hyper data.

We can also use `plot.scanone` to plot both the interval mapping results and those by marker regression (obtained in the previous section) together in one figure. This may be done in a couple of different ways. First, we can send both results to the `plot.scanone` function. We plot the results for chromosomes 4 and 12 as follows; see Fig. 4.5.

```
> plot(out.em, out.mr, chr=c(4, 12), col=c("blue", "red"),
+      ylab="LOD score")
```

We can produce the same figure by first plotting the interval mapping results and then adding the marker regression results using `add=TRUE`.

```
> plot(out.em, chr=c(4, 12), col="blue", ylab="LOD score")
> plot(out.mr, chr=c(4, 12), col="red", add=TRUE)
```

Finally, we can create a black-and-white plot with different line types, using the `lty` argument. Use `lty=1` for a solid line, `lty=2` for a dashed line, and `lty=3` for a dotted line. We give it a vector with two plot types; the first line type is used for the first result sent to the plot; the second line type is for the second result.

```
> plot(out.em, out.mr, chr=c(4, 12), col="black", lty=1:2,
+      ylab="LOD score")
```

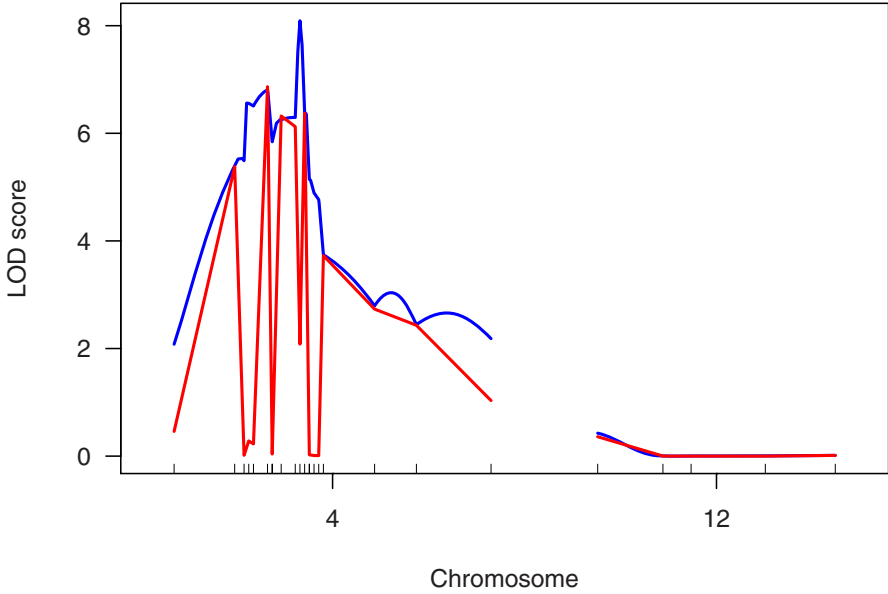


Figure 4.5. LOD scores for selected chromosomes for the **hyper** data by standard interval mapping (blue) and marker regression (red).

4.2.2 Haley–Knott regression

Haley–Knott regression provides a fast approximation of the results of standard interval mapping. In standard interval mapping, we assume that $y_i|g_i \sim N(\mu_{g_i}, \sigma^2)$, where y_i is the phenotype of individual i and g_i is its (unobserved) QTL genotype. We further calculate $p_{ij} = \Pr(g_i = j|\mathbf{M}_i)$, where $\mathbf{M}_i$ is the marker genotype data for individual i . Recall that $y_i|\mathbf{M}_i$ follows a mixture of normal distributions.

Note that $E(y_i|\mathbf{M}_i) = \sum_j p_{ij}\mu_j$ and so the conditional phenotype average, given the available marker data, is linear in the μ_j . This suggests that the μ_j might be estimated by linear regression of the y_i on the p_{ij} .

This is Haley–Knott regression. At each position on our grid across the genome, we calculate the p_{ij} and then regress the phenotype on this matrix. In doing so, we pretend that $y_i|\mathbf{M}_i \sim N(\sum_j p_{ij}\mu_j, \sigma^2)$. That is, we replace the normal mixture with a single normal distribution, though with the correct mean function. We may thus calculate a LOD score as

$$\text{LOD} = \frac{n}{2} \log_{10} \left(\frac{\text{RSS}_0}{\text{RSS}_1} \right)$$

where RSS_0 is the null residual sum of squares and RSS_1 is the residual sum of squares from the regression of the y_i on the p_{ij} .

Haley–Knott regression can be much faster than standard interval mapping, as an iterative algorithm is not needed; one performs a single regression

at each position. However, the treatment of missing genotype information is less than ideal, and so its approximation of standard interval mapping can be poor in regions of low genotype information (such as widely spaced or incompletely genotyped markers). The approximation is especially poor in the case of selective genotyping (in which only individuals with extreme phenotypes are genotyped). We will discuss this issue further in Sec. 4.2.5, below.

Example

To perform Haley–Knott regression, we again need the genotype probabilities calculated by `calc.genoprob`, though we do not need to run the function again here, as it was executed in order to get the results by standard interval mapping. We again use the `scanone` function, and use `method="hk"` for Haley–Knott regression, as follows.

```
> out.hk <- scanone(hyper, method="hk")
```

We plot the results with those of standard interval mapping with the following. We look only at chromosomes 1, 4, and 15, so that the differences may be more clearly seen; the result appears in Fig. 4.6.

```
> plot(out.em, out.hk, chr=c(1,4,15), col=c("blue","red"),
+       ylab="LOD score")
```

The results from standard interval mapping and Haley–Knott regression are seen to be quite similar, though they deviate from each other at the ends of the chromosomes, particularly at the distal end of chromosome 15. The discrepancies occur in regions of missing genotype information, and are particularly affected by the selective genotyping strategy used for these data. The terminal markers on chromosomes 1 and 4, and the distal markers on chromosome 15, were genotyped at only the 92 individuals with most extreme phenotypes.

R/qtl contains a function `-.scanone` for subtracting two sets of LOD scores from each other, provided that they conform exactly (that they come from the same cross and that calculations were performed at the same density). We thus may look at the differences in the LOD scores from the two methods as follows. The result appears in Fig. 4.7.

```
> plot(out.hk - out.em, chr=c(1,4,15), ylim=c(-0.5, 1.0),
+       ylab=expression(LOD[HK] - LOD[EM]))
> abline(h=0, lty=3)
```

We used `abline` to add a dotted horizontal line at 0, and we used the function `expression` to get a fancy *y*-axis label. Type `?plotmath` to read about the possibilities with `expression`, and consider the following code. (We omit the resulting figure and any explanation; hopefully the interested reader can figure this out.)

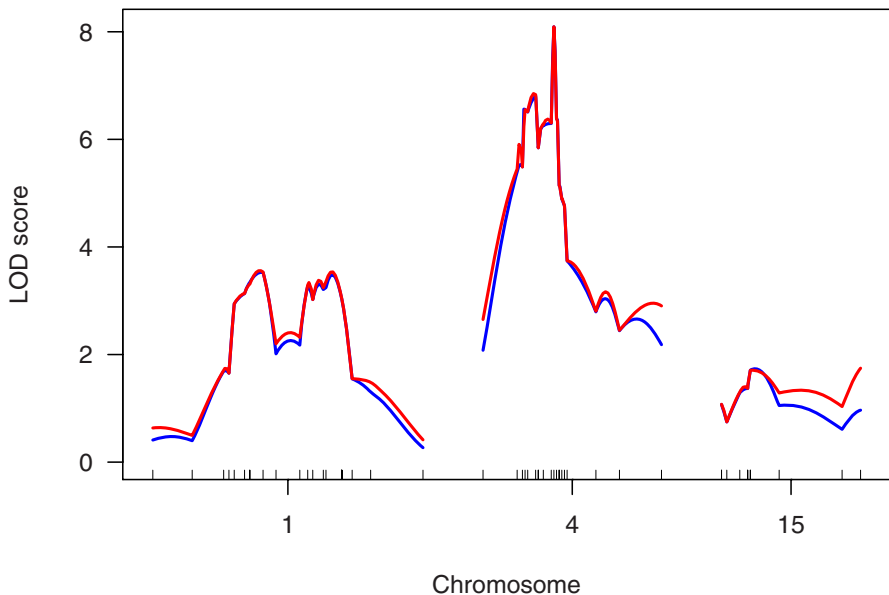


Figure 4.6. LOD scores for selected chromosomes for the **hyper** data by standard interval mapping (blue) and Haley–Knott regression (red).

```
> plot(rnorm(100), rnorm(100), xlab=expression(hat(mu)[0]),
+      ylab=expression(alpha^beta),
+      main=expression(paste("Plot of ", alpha^beta,
+      " versus ", hat(mu)[0])))
```

4.2.3 Extended Haley–Knott regression

An improved version of Haley–Knott regression may be obtained by also considering the variances. In Haley–Knott regression, we used the fact that $E(y_i|\mathbf{M}_i) = \sum_j p_{ij}\mu_j$ and made the approximation $y_i|\mathbf{M}_i \sim N(\sum_j p_{ij}\mu_j, \sigma^2)$. That is, we approximate the mixture distribution with a single normal distribution with the correct mean, but with constant variance independent of genotype.

In the extended Haley–Knott regression method, we note that

$$\begin{aligned} \text{var}(y_i|\mathbf{M}_i) &= \text{var}[E(y_i|g_i)|\mathbf{M}_i] + E[\text{var}(y_i|g_i)|\mathbf{M}_i] \\ &= \text{var}(\mu_{g_i}|\mathbf{M}_i) + E(\sigma^2|\mathbf{M}_i) \\ &= \sum_j p_{ij}[\mu_j - \sum_k p_{ik}\mu_k]^2 + \sigma^2 \end{aligned}$$

Write $m_i(\boldsymbol{\mu}) = \sum_j p_{ij}\mu_j$ and $v_i(\boldsymbol{\mu}, \sigma^2) = \sum_j p_{ij}[\mu_j - m_i(\boldsymbol{\mu})]^2 + \sigma^2 = \sum_j p_{ij}\mu_j^2 - (\sum_j p_{ij}\mu_j)^2 + \sigma^2$. In the extended Haley–Knott method, we assume

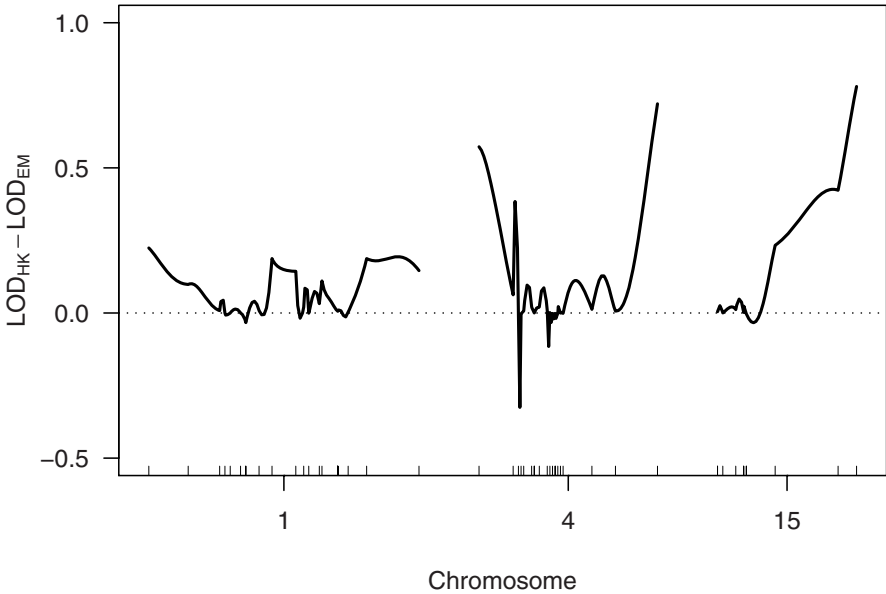


Figure 4.7. Differences in the LOD scores from Haley–Knott regression and standard interval mapping for selected chromosomes from the `hyper` data.

that $y_i | \mathbf{M}_i \sim N[m_i(\boldsymbol{\mu}), v_i(\boldsymbol{\mu}, \sigma^2)]$. That is, we replace the mixture distribution with a single normal distribution but with the correct mean and variance functions.

We estimate the μ_j and σ^2 by maximum likelihood (though with this approximate normal model). This requires an iterative method, though it generally converges more quickly than the EM algorithm under the mixture model.

The extended Haley–Knott method is not as fast as Haley–Knott regression, but it provides an improved approximation and is still somewhat faster than standard interval mapping. Most importantly, the extended Haley–Knott method is more robust than standard interval mapping; see Sec. 4.2.5.

Example

As with standard interval mapping and Haley–Knott regression, we need the genotype probabilities calculated by `calc.genoprob`, though we do not need to run the function again here. We use the `scanone` function with `method="ehk"` for the extended Haley–Knott method, as follows.

```
> out.ehk <- scanone(hyper, method="ehk")
```

We can plot the three interval mapping methods together with the following. The results appear in Fig. 4.8.

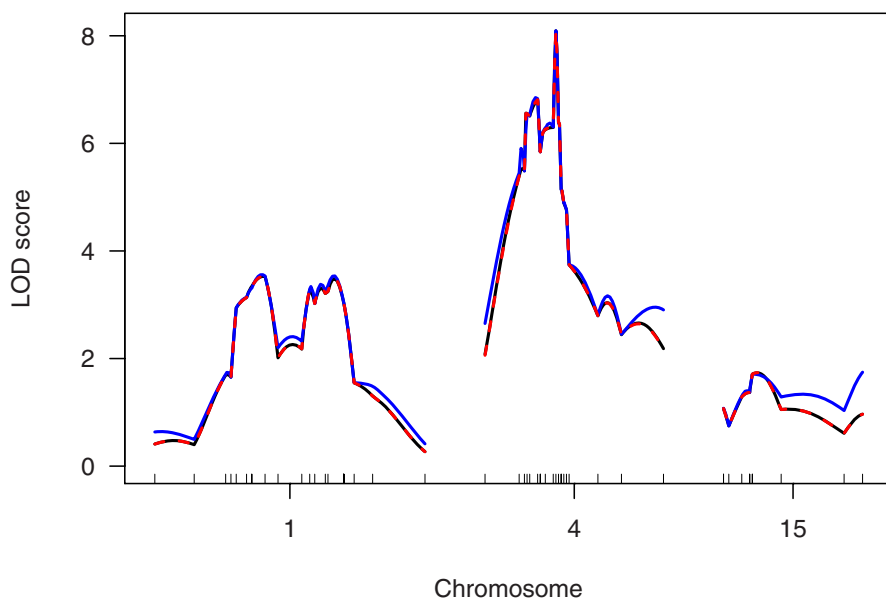


Figure 4.8. LOD scores for selected chromosomes for the `hyper` data by standard interval mapping (black), Haley-Knott regression (blue), and the extended Haley-Knott method (red, dashed).

```
> plot(out.em, out.hk, out.ehk, chr=c(1,4,15), ylab="LOD score",
+      lty=c(1,1,2))
```

The colors black, blue, and red are used by default. The argument `lty` is used to define line types (1 is solid; 2 is dashed). As we will see, the results by standard interval mapping and the extended Haley-Knott method are almost indistinguishable; we plot the extended Haley-Knott results with dashed curves so that the interval mapping results may still be seen.

Note how much more closely the results from the extended Haley-Knott method follow the results of standard interval mapping. The black curves are completely covered by the red curves, as standard interval mapping and the extended Haley-Knott method give results that are almost indistinguishable.) Up to three results may be plotted with a single call to `plot.scanone`. Alternatively, we may use `add=TRUE`, to obtain this plot.

```
> plot(out.em, chr=c(1,4,15), ylab="LOD score")
> plot(out.hk, chr=c(1,4,15), col="blue", add=TRUE)
> plot(out.ehk, chr=c(1,4,15), col="red", lty=2, add=TRUE)
```

To more clearly see the differences among the results, we can again plot the differences between the LOD scores. The results appear in Fig. 4.9.

```
> plot(out.hk - out.em, out.ehk - out.em, chr=c(1, 4, 15),
+      col=c("blue", "red"), ylim=c(-0.5, 1),
```

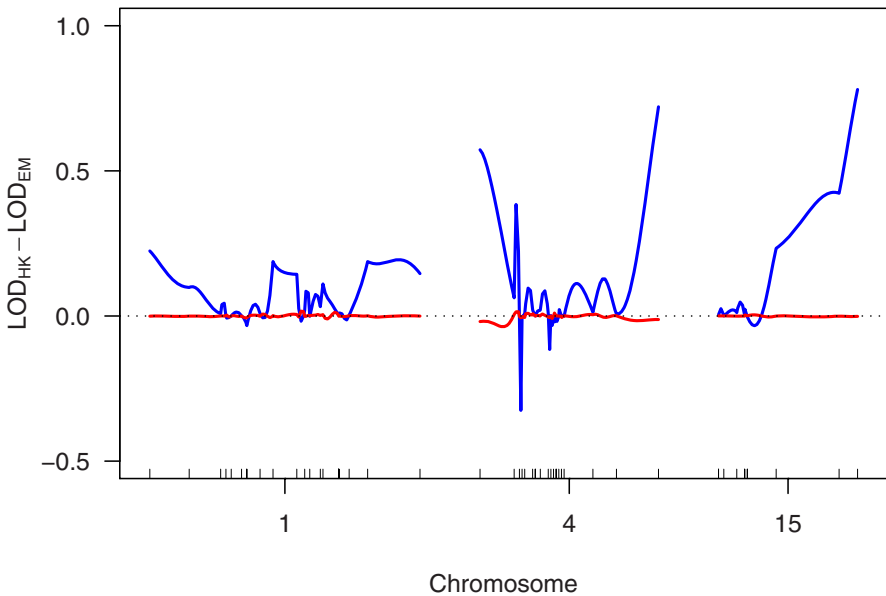


Figure 4.9. Differences in the LOD scores from Haley–Knott regression (in blue) and the extended Haley–Knott method (in red) from the LOD scores of standard interval mapping, for selected chromosomes from the `hyper` data.

```
+      ylab=expression(LOD[HK] - LOD[EM]))
> abline(h=0, lty=3)
```

4.2.4 Multiple imputation

As we discussed in Sect. 1.3, the QTL mapping problem can be split into two parts: the missing data problem and the model selection problem. The multiple imputation approach dispenses with the missing data problem by filling in all missing genotype data, even at sites between markers (on a grid along the chromosomes). With complete genotype data, the fit of a QTL model reduces to ANOVA (for single-QTL models) or multiple regression (for multiple-QTL models).

The only wrinkle is that such imputations must be done multiple times, with the final result being a combination of the results from the multiple imputations. Moreover, the combination of the single-imputation results can be complicated. The imputations are simple and model fit with each imputation is simple, but the combination of the imputations can be difficult.

Genotypes are imputed randomly, but conditional on the observed marker genotype data: we simulate from the joint genotype distribution given the observed data. For example, consider Fig. 4.10, which illustrates the imputation of a single backcross individual's genotype data. The observed genotype data

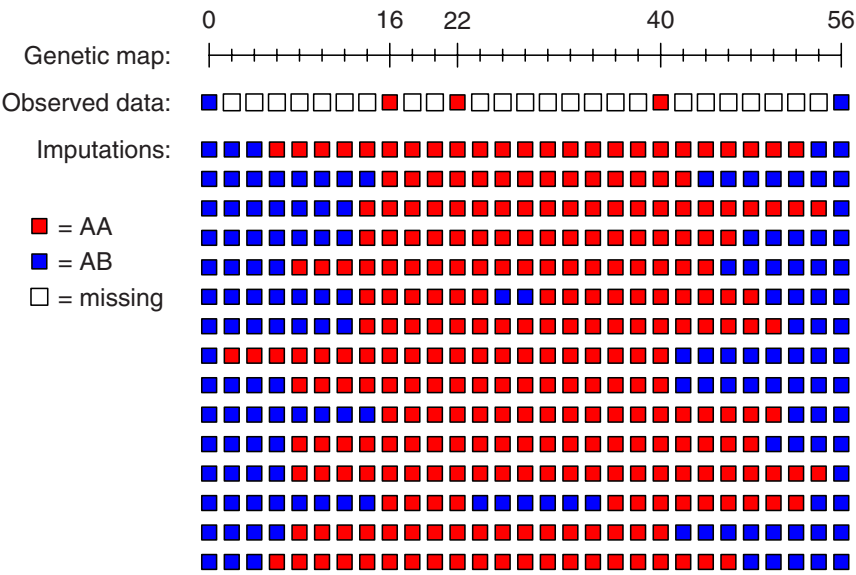


Figure 4.10. Illustration of multiple imputations for a single backcross individual. Red and blue squares correspond to homozygous and heterozygous genotypes, respectively, while open squares indicate missing data. The marker genotype data for the individual is shown at the top, below a genetic map (in cM) for the chromosome; multiple imputations of the genotype data, at the markers and at intervening positions at 2 cM steps along the chromosome, are shown below.

at five genetic markers is shown at the top, followed by the imputed genotypes for 15 different imputations. Note that the positions of the recombination events vary among the imputations, and a couple of imputations exhibit double-crossovers between markers. The imputed genotypes at the markers match those observed, as these data were simulated assuming no genotyping errors (an assumption that may be relaxed).

Such multiple imputations would be obtained on all individuals. With a given set of imputed data, we can simply perform a t test or ANOVA at each position, as with such complete genotype data on all individuals, we know how to split the individuals into genotype groups. Following tradition, we express the results as LOD scores.

As a further illustration, consider Fig. 4.11, which contains imputation results for chromosome 4 of the `hyper` data. The LOD curve from each of 16 imputations are shown in gray, and a combined LOD curve from a total of 64 imputations is shown in black. At markers with complete genotype data, all LOD curves coincide, as all imputations give the same set of data. In regions with less genotype information, the LOD curves from the individual imputations deviate from one another.

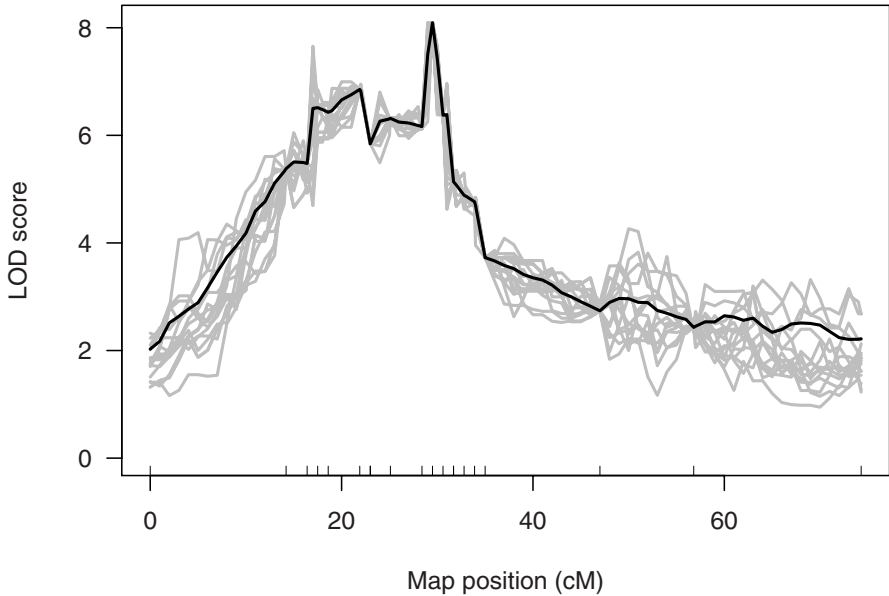


Figure 4.11. Illustration of the imputation results for chromosome 4 of the **hyper** data. The individual LOD curves from 16 imputations are shown in gray; the combined LOD score from a total of 64 imputations is shown in black.

For the normal model, the combined LOD score is the average of the LOD scores from the individual imputations, though on the 10^{LOD} scale. To obtain a more stable estimate of this average, we use a trimmed mean. In the case of m imputations, we trim the lowest and highest $\log_2(m)/2$ LOD scores. (More precisely, we trim $\lfloor \log_2(m)/2 \rfloor$ from each end, where $\lfloor x \rfloor$ is the greatest integer $\leq x$. We generally take the number of imputations to be a power of 2, like 64 or 128: $m = 2^k$ for some positive integer k .) Moreover, we assume that the LOD scores at a particular position, across imputations, approximately follow a lognormal distribution, and we use the fact that, if $L = \ln(W) \sim N(\mu, \sigma^2)$, so that W is lognormally distributed, $E(W) = \exp(\mu + \sigma^2/2)$.

To be precise, the combined LOD scores that we calculate by the multiple imputation method are not truly LOD scores. Strictly speaking, this is a Bayesian method, and the combined scores are the log posterior distribution (LPD) of QTL location, but the results are generally similar to the LOD scores from standard interval mapping.

The multiple imputation approach is intensive in both computation time and memory use. While it is more robust than standard interval mapping, it has little advantage over the extended Haley–Knott method for single-QTL models, particularly because of the large up-front cost to obtain the imputations. The multiple imputation approach has greatest value for the fit and exploration of multiple-QTL models (see Chap. 9).

Example

To perform multiple imputation in R/qtl, we first perform the imputations using the `sim.geno` function. This function is similar to `calc.genoprob`, though it has an additional argument, `n.draws`, through which the number of imputations is specified.

```
> hyper <- sim.geno(hyper, step=1, n.draws=64, error.prob=0.001)
```

The imputed genotypes can take up an *enormous* amount of memory, especially if `step` is small, `n.draws` is large, and there are many individuals in the cross. In such cases, you will need a computer with a lot of RAM. You may wish to perform initial analyses with a rather coarse `step` size and with fewer imputations, reserving the refined `step` and larger `n.draws` for the fit of later multiple-QTL models (see Chap. 9), in which case you can focus on just those chromosomes that appear to harbor a QTL. If time and memory are not as issue, more imputations are always better (just as a smaller `step` size is always better). With relatively complete genotype data and relatively dense markers, very few imputations will be required; the more sparse the genotype information, the more imputations should be used. Repeating the analysis with independent imputations can give a good indication of the need for a larger number of imputations. If the results are hardly distinguishable, the chosen number of imputations is sufficient.

The analysis is again accomplished with `scanone`, with `method="imp"` for the multiple imputation method. For example, we type the following to perform interval mapping by multiple imputation with the `hyper` data.

```
> out.imp <- scanone(hyper, method="imp")
```

We plot the results for selected chromosomes with those from standard interval mapping (Sec. 4.2.1) via the following; the results appear in Fig. 4.12.

```
> plot(out.em, out.imp, chr=c(1,4,15), col=c("blue", "red"),
+      ylab="LOD score")
```

It is again worthwhile to look at the differences between the LOD curves. See Fig. 4.13.

```
> plot(out.imp - out.em, chr=c(1,4,15), ylim=c(-0.5, 0.5),
+      ylab=expression(LOD[IMP] - LOD[EM]))
> abline(h=0, lty=3)
```

4.2.5 Comparison of methods

In this section, we summarize the relative advantages and disadvantages of the various single-QTL mapping methods that we have discussed in this chapter. For a quick summary, see Table 4.2 on page 102.

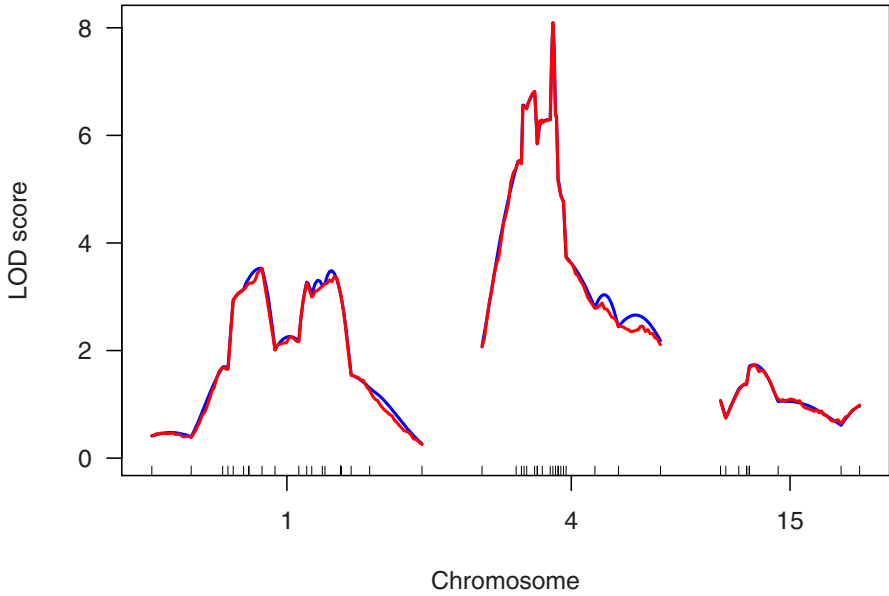


Figure 4.12. LOD scores for selected chromosomes for the *hyper* data by standard interval mapping (blue) and multiple imputation (red).

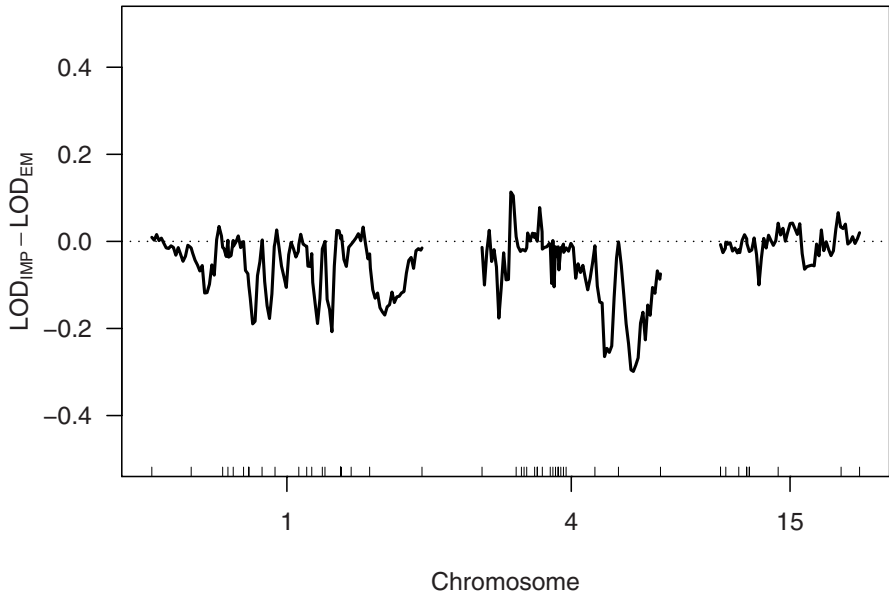


Figure 4.13. Differences in the LOD scores from multiple imputation and standard interval mapping for selected chromosomes from the *hyper* data.

Marker regression is not recommended for use in practice, except in the case of dense markers with complete genotype data (as is sometimes the case with data on recombinant inbred lines), because individuals with missing genotype at a marker must be omitted from the analysis, and one cannot inspect positions between markers. The interval mapping methods take account of missing genotype data.

Standard interval mapping, which uses maximum likelihood estimation in a normal mixture model, is generally the preferred method, but it can give artifacts. Recall that the LOD score is the $\log_{10}$ likelihood ratio, comparing the hypothesis of a single QTL at the position under test to the null hypothesis of no QTL anywhere in the genome. If the phenotype distribution exhibits multiple modes, so that it would be better approximated by a mixture of normal distributions rather than a single normal distribution, one can get spuriously large LOD scores in regions of low genotype information (such as a large gap between typed markers). At a typed marker, one is constrained by the observed genotypes, but in a region with low genotype information, the likelihood under the alternative essentially concerns the fit of a mixture model.

For example, consider the `listeria` data. The phenotype is time-to-death following infection with *Listeria monocytogenes*, but a large number of individuals recovered from infection, and so their phenotype was censored (and recorded as 264 hours); see Fig. 2.7 on page 35. The markers are relatively dense, and the genotype data relatively complete, and so application of standard interval mapping with these data do not cause problems. If we omit most of the markers on a chromosome, we can illustrate our point. Chromosome 1 was typed at 13 markers; let us omit all but the terminal markers. We use the function `markernames` to pull out the marker names for chromosome 1 and `drop.markers`, to drop all but the first and last markers.

```
> data(listeria)
> mar2drop <- markernames(listeria, chr=1)[2:12]
> listeria <- drop.markers(listeria, mar2drop)
```

If we now perform standard interval mapping, we will see a large LOD peak on chromosome 1, in the middle of the large interval with no genotype data (see Fig. 4.14). The other interval mapping methods do not exhibit this problem; we consider just the Haley–Knott methods here. We use the argument `chr=1` in `scanone` so that only chromosome 1 is analyzed.

```
> listeria <- calc.genoprob(listeria, step=1, error.prob=0.001)
> outl.em <- scanone(listeria, chr=1)
> outl.hk <- scanone(listeria, chr=1, method="hk")
> outl.ehk <- scanone(listeria, chr=1, method="ehk")
> plot(outl.em, outl.hk, outl.ehk, ylab="LOD score",
+      lty=c(1,1,2))
```

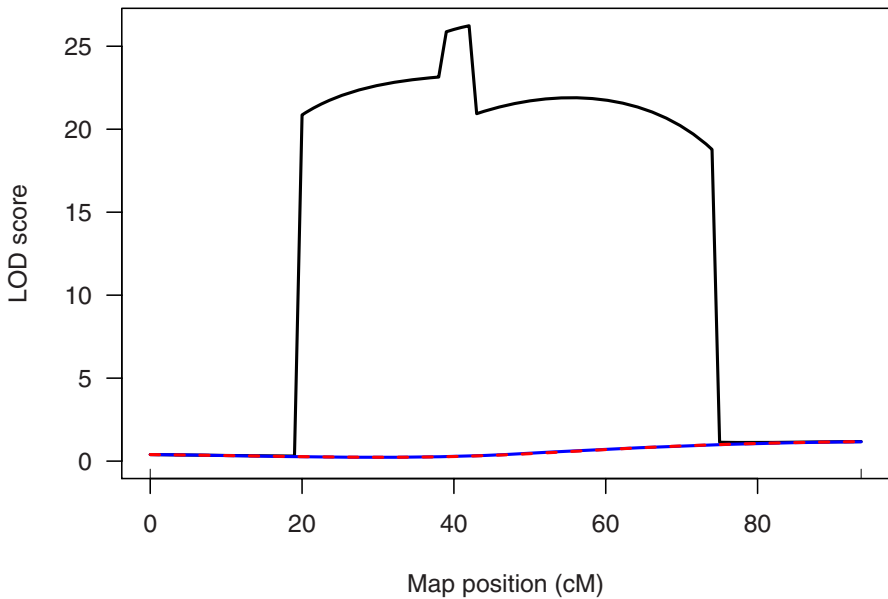


Figure 4.14. LOD scores by standard interval mapping (black), Haley–Knott regression (blue) and the extended Haley–Knott method (red, dashed) for chromosome 1 of the *listeria* data, when the genotype data for all but the terminal markers have been omitted.

The discontinuities in the LOD curve from standard interval mapping concern multiple modes in the likelihood surface. At a typed marker, one is constrained by the observed genotype data. At the center of the large interval, there is essentially no genotype data, and so one is fitting a pure mixture model.

The Haley–Knott method is more robust than standard interval mapping; however, the approximation used in Haley–Knott regression, in which we regress the phenotype on conditional genotype probabilities, can be poorly behaved with appreciable missing genotype data, particularly in the case of selective genotyping. In the selective genotyping strategy (discussed further in Chap. 6), only individuals with extreme phenotypes (for example, the individuals in the upper and lower 10% of the phenotype distribution) are genotyped. In the case of an inexpensive phenotype, this greatly reduces the cost of a study yet provides nearly equivalent power for QTL detection.

Consider a backcross with genotypes coded as 0 and 1. In Haley–Knott regression, an individual with no genotype data will be treated as if its genotype were $1/2$ (and were known to be $1/2$). This results in a slightly inflated estimate of the effect of a QTL but also a greatly reduced estimate of the residual variation, and so can give inflated LOD scores.

In the extended Haley–Knott method, individuals with no genotype data are also treated as having genotype 1/2, but their residual variance is adjusted, so they are given very little weight in the estimation. Standard interval mapping and the multiple imputation method explicitly take account of the fact that the individuals without genotypes have an equal chance of being homozygous and heterozygous. As a result, these other methods give remarkably similar LOD curves, irrespective of whether the individuals without genotype data are included in the analysis.

If one omitted individuals with absolutely no genotype data, Haley–Knott regression will provide a good approximation to standard interval mapping. However, often (as with the **hyper** data) selective genotyping is followed by complete genotyping at certain markers, and so no individual is completely lacking in genotype data.

Consider the **hyper** data as an example. The relationship between blood pressure and the genotype at marker D4Mit214 is shown in the left panel Fig. 4.15. The regression line of phenotype on genotype (equivalent to a t test) is shown. In the right panel, the genotype data for all but the 92 individuals with extreme phenotypes have been omitted. The blue line is the regression line when only the genotyped individuals are considered. The red line is the regression line obtained when those not genotyped are placed intermediate between the two genotyped groups, as is done in Haley–Knott regression. The regression lines in the right panel are more steep than that in the left panel, but the biggest difference is that, in the right panel, with the ungenotyped individuals placed in the center, the variation around the regression line is artificially reduced.

Let us look more closely at the LOD curves that will be obtained by the different methods. In the **hyper** data, further genotyping was performed in regions showing initial evidence for a QTL; we will omit these genotypes, to convert these data to the “pure” selective genotyping form.

First, we identify the markers that have mostly missing data (those that were typed only on recombinant individuals) and use **drop.markers** to remove them from the data set. We assign the revised data to a new object, **hyper.rev**.

```
> data(hyper)
> nt.mar <- ntyped(hyper, "mar")
> mar2drop <- names(nt.mar[nt.mar < 92])
> hyper.rev <- drop.markers(hyper, mar2drop)
```

Next, we eliminate the genotype data for the individuals with intermediate phenotype, who may be identified by their missing genotype data. This requires a loop over the chromosomes; the genotype data for the relevant individuals is replaced with NA (for “missing”).

```
> nm.ind <- nmissing(hyper.rev)
> ind2drop <- nm.ind > 0
```

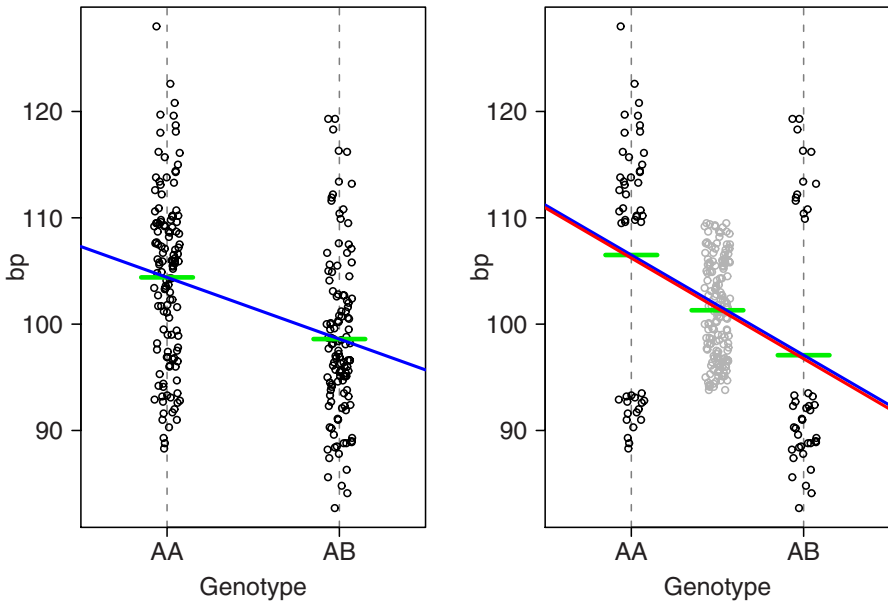


Figure 4.15. Relationship between blood pressure and the genotype at D4Mit214 for the *hyper* data. The horizontal green segments indicate the within-group averages. In the left panel, the data for all individuals is shown with the regression line of phenotype on genotype. In the right panel, the genotype data for all but the 92 individuals with extreme phenotypes is omitted. The blue line is the regression line obtained with only the 92 phenotyped individuals. The red line comes from a regression with all individuals, but with those not genotyped placed intermediate between the two genotype groups, as is done in Haley-Knott regression.

```
> for(i in 1:nchr(hyper))
+   hyper.rev$geno[[i]]$data[ind2drop,] <- NA
```

Now, we perform genome scans using all individuals. We will use each of standard interval mapping, Haley-Knott regression and the extended Haley-Knott method.

```
> hyper.rev <- calc.genoprob(hyper.rev, step=1,
+                             error.prob=0.001)
> out1.em <- scanone(hyper.rev)
> out1.hk <- scanone(hyper.rev, method="hk")
> out1.ehk <- scanone(hyper.rev, method="ehk")
```

The LOD curves from the three methods are shown in Fig. 4.16, which was created as follows.

```
> plot(out1.em, out1.hk, out1.ehk, ylab="LOD score",
+       lty=c(1,1,2))
```

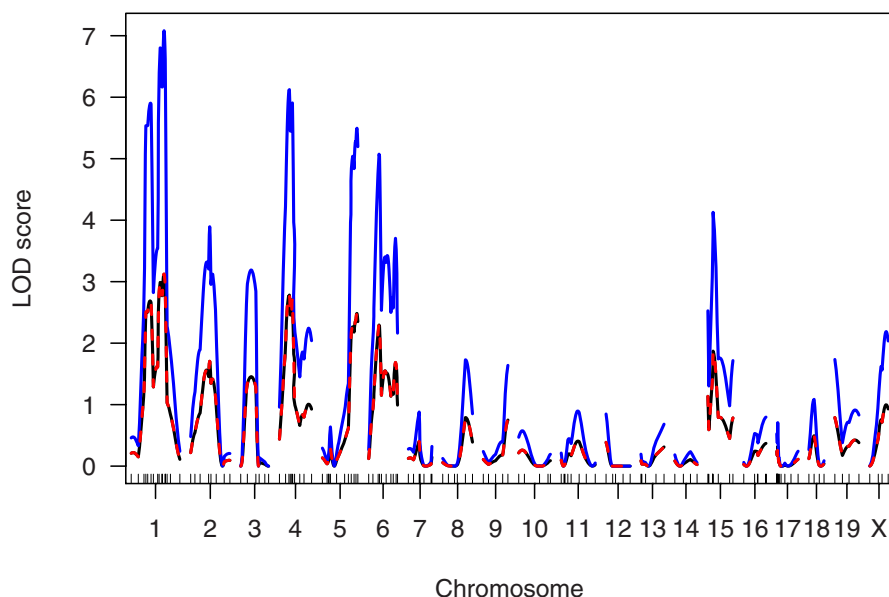


Figure 4.16. LOD curves from standard interval mapping (black), Haley–Knott regression (blue) and the extended Haley–Knott method (red, dashed), for the **hyper** data when all individuals are considered but genotype data for individuals with intermediate phenotype are omitted.

The LOD scores from Haley–Knott regression (in blue) are inflated. The results of standard interval mapping (black) and the extended Haley–Knott method (red) almost completely overlap. Thus, to distinguish among the methods more clearly, we plot the differences between the LOD scores for the Haley–Knott methods and those from standard interval mapping.

```
> plot(out1.hk-out1.em, out1.ehk-out1.em, col=c("blue", "red"),
+      ylim=c(-0.1, 4), ylab=expression(LOD[HK] - LOD[EM]))
> abline(h=0, lty=3)
```

The plot of the differences, in Fig. 4.17, confirm that the extended Haley–Knott method gives results that are virtually identical to standard interval mapping.

We now perform the same analyses, considering only the genotyped individuals. We use `subset.cross` to create a new version of the data with only the genotyped individuals. The vector `ind2drop` was created above to contain the logical values `TRUE` and `FALSE`, with `TRUE` indicating individuals with no genotype data. We use `!ind2drop` to reverse the `TRUE/FALSE` values (`!` is the logical “not”), to pull out just those individuals with genotype data.

```
> hyper.ex <- subset(hyper.rev, ind = !ind2drop)
> out2.em <- scanone(hyper.ex)
```

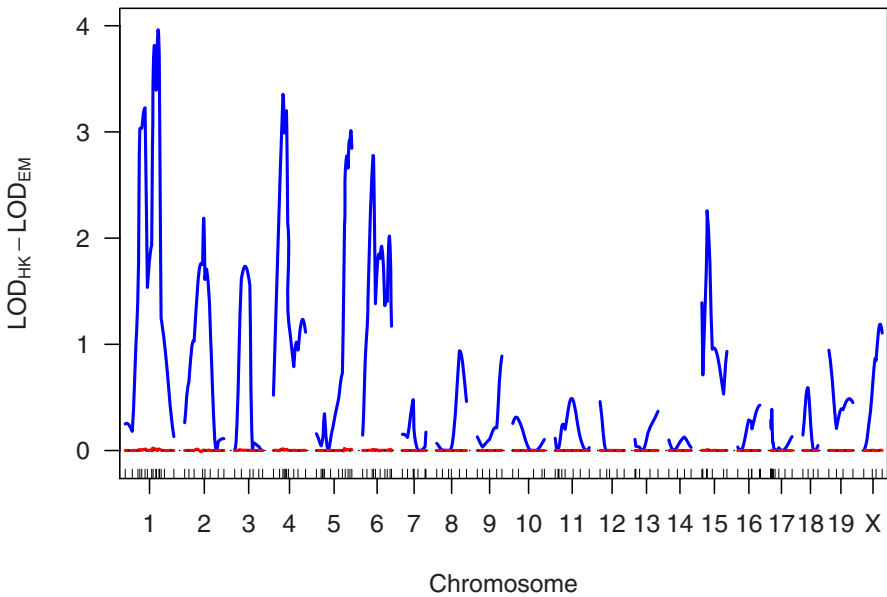


Figure 4.17. Differences in the LOD curves of Haley–Knott regression (blue) and the extended Haley–Knott method (red), from the LOD curves of standard interval mapping, for the `hyper` data when all individuals are considered but genotype data for individuals with intermediate phenotype are omitted.

```
> out2.hk <- scanone(hyper.ex, method="hk")
> out2.ehk <- scanone(hyper.ex, method="ehk")
```

Let us just look at the differences in the LOD scores for the methods for this case (with only genotyped individuals considered) and the LOD curves from standard interval mapping when all individuals were considered; see Fig. 4.18.

```
> plot(out2.em-out1.em, out2.hk-out1.em, out2.ehk-out1.em,
+      ylim=c(-0.1, 0.2), col=c("black", "green", "red"),
+      lty=c(1,3,2), ylab="Difference in LOD scores")
> abline(h=0, lty=3)
```

The methods all give similar results, and the results are not too different from standard interval mapping when all individuals are considered (but with the genotypes of the individuals with intermediate phenotype omitted).

In summary, standard interval mapping can give spuriously large LOD scores in regions of low genotype information, if the phenotype distribution is better approximated by a normal mixture than by a single normal distribution; the other methods do not have this problem. Haley–Knott regression can provide a poor approximation to interval mapping in the case of low genotype information, and can give inflated LOD scores in the presence of selective

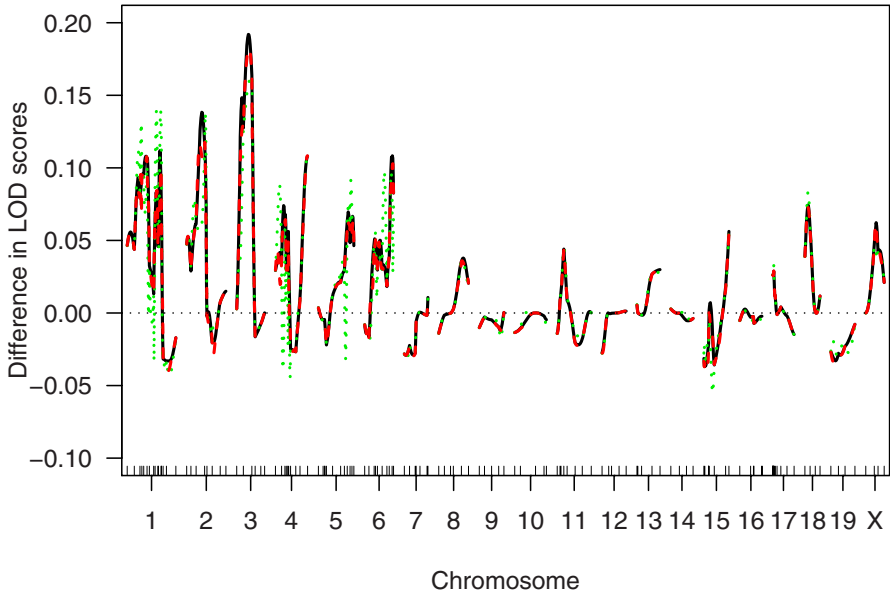


Figure 4.18. Differences in the LOD curves (for the `hyper` data) from standard interval mapping (in black), Haley–Knott regression (in green, dotted) and the extended Haley–Knott method (in red, dashed), all calculated with only the data on the individuals with extreme phenotypes, from the LOD curves of standard interval mapping, with all individuals but with genotype data for individuals with intermediate phenotype omitted.

Table 4.2. Relative advantages and disadvantages of the four interval mapping methods.

	Use of genotype information	Robustness	Selective genotyping	Speed
Standard interval mapping	++	–	+	–
Haley–Knott	–	+	–	+
Extended Haley–Knott	+	+	+	–
Multiple imputation	++	+	+	--

genotyping. The extended Haley–Knott method may be preferred as it suffers from neither of these problems. These points are summarized in Table 4.2.

Our last point concerns computation time. Haley–Knott regression is especially fast, and the extended Haley–Knott method is intermediate between Haley–Knott regression and standard interval mapping in computation time. The multiple imputation approach is particularly slow for the fit of single-QTL models, and is best reserved for the fit of multiple-QTL models.

We can measure computation time with the `system.time` function. The following times were obtained on a Mac Pro. We will consider the `hyper` data.

First, look at the time to run `calc.genoprob`. We use `step=0.25` so that the later comparison of the methods is most clear.

```
> data(hyper)
> system.time(hyper <- calc.genoprob(hyper, step=0.25,
+                                   error.prob=0.001))

   user  system elapsed 
1.392   0.472   1.882
```

In the output from `system.time`, the first number is the CPU time (in seconds) and the third number is the total time.

Now let us look at the four interval mapping methods.

```
> system.time(test.hk <- scanone(hyper, method="hk"))

   user  system elapsed 
0.238   0.095   0.337

> system.time(test.ehk <- scanone(hyper, method="ehk"))

   user  system elapsed 
0.982   0.093   1.085

> system.time(test.em <- scanone(hyper, method="em"))

   user  system elapsed 
1.105   0.094   1.207
```

The extended Haley–Knott method is intermediate between the other two methods, though here it is not much faster than standard interval mapping.

For the multiple imputation approach, the calculations (in `sim.geno` and `scanone` with `method="imp"`) scale linearly in `n.draws`.

```
> system.time(hyper <- sim.geno(hyper, step=0.25,
+                               error.prob=0.001, n.draws=32))

   user  system elapsed 
11.795   4.253  16.288

> system.time(test.imp <- scanone(hyper, method="imp"))

   user  system elapsed 
3.002   0.608   3.706
```

The imputations themselves are quite time consuming, and the actual analysis is also slower than the EM algorithm. (This will depend on the number of iterations needed for convergence of EM; the time for one iteration of EM is approximately the same as the time analyzing one imputation.)

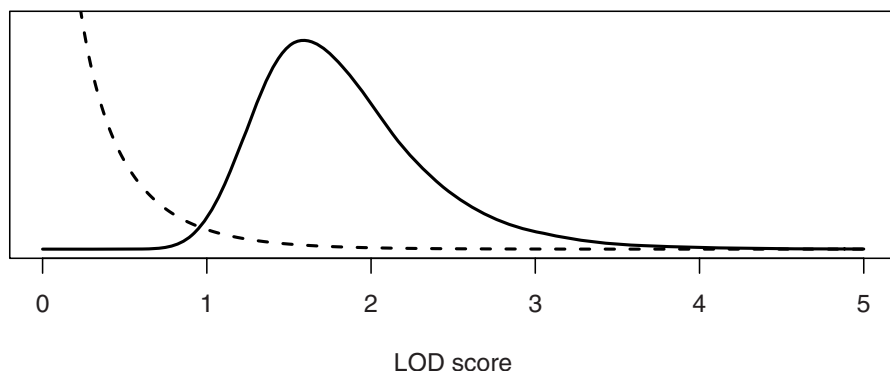


Figure 4.19. Approximate null distribution of the LOD score at a particular position (dashed curve) and of the maximum LOD score genome-wide (solid curve) for a backcross in the mouse.

4.3 Significance thresholds

A LOD score indicates evidence for the presence of a QTL, with larger LOD scores corresponding to greater evidence. The question is, how large is large? In answering this question, we must take account of the genome-wide search for QTL.

We consider the global null hypothesis, that there is no QTL anywhere in the genome. (This is almost always clearly false, as we generally start with inbred lines that show a clear difference in the phenotype, which implies that there must be QTL. Nevertheless, the procedures described here are useful.) Rather than consider the null distribution of the LOD score at a particular position (the dashed curve in Fig. 4.19), we consider the null distribution of the genome-wide maximum LOD score (the solid curve in Fig. 4.19).

As seen in Fig. 4.19, in a backcross with no QTL, at any fixed position in the genome, one will typically see a LOD score of about 0.25, and will seldom see a LOD score greater than 1. However, one will typically see a LOD score > 1.5 *somewhere* in the genome, and will see a LOD score > 2.5 about 10% of the time. We compare our observed LOD scores to the distribution of the genome-wide maximum LOD score, in the case that there were no QTL anywhere. The 95th percentile of this distribution may be used as a genome-wide LOD threshold. Alternatively, one may calculate a genome-scan-adjusted p -value corresponding to an observed LOD score: the chance, under the null hypothesis of no QTL, of obtaining a LOD score that large or larger somewhere in the genome.

The null distribution of the genome-wide maximum LOD depends on a number of factors, including the type of cross (backcross or intercross), the size of the genome (in cM), the number of individuals, the number of typed markers, the pattern of missing genotype data, and the phenotype distribution.

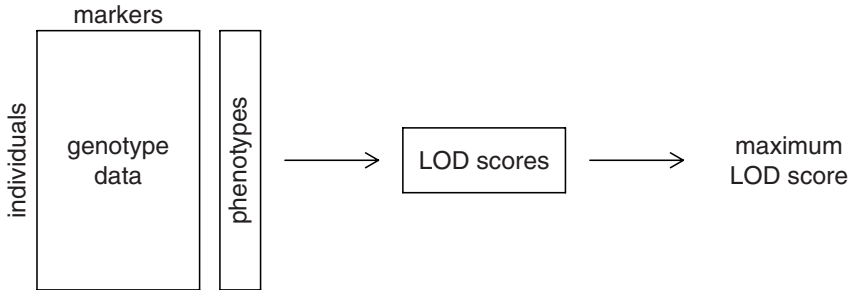


Figure 4.20. Diagram of the interval mapping process.

The type of cross and the genome size have the greatest influence. One may derive this null distribution by several methods: computer simulation, analytic calculations (e.g., for the case of infinitely dense markers), or by a permutation test. We prefer permutation tests, as they best account for the particular features of one's data.

The process involved in interval mapping is illustrated in Fig. 4.20. The data consist of a rectangle of marker genotype data plus a column of phenotypes. QTL analysis results in a set of LOD curves, and one immediately notes the genome-wide maximum LOD curve. The question is, if there were no QTL (so that there was no association between phenotypes and the genotypes), what sort of maximum LOD scores might be obtained?

In a permutation test, we tackle this directly by permuting (i.e., randomizing or shuffling) the phenotypes relative to the genotype data. That is, the genotype data rectangle remains intact, and the observed phenotypes are kept the same, but which phenotype corresponds to which genotypes is randomized. We apply our QTL mapping method to this shuffled version of the data and obtain a set of LOD curves; we then derive the genome-wide maximum LOD score. The process is repeated a number of times, which results in a set of maximum LOD scores, $M_1^*, M_2^*, \dots, M_r^*$, where r is the number of permutation replicates (generally $r = 1000$ or $10,000$). We may use the 95th percentile of the M_i^* as an estimated genome-wide LOD threshold, or we may calculate a genome-scan-adjusted p -value as the proportion of the M_i^* that meet or exceed a particular observed LOD score.

It should be noted that, in the case of selectively genotyped data, the usual permutation test is not appropriate, as the individuals are no longer exchangeable. One should instead use a *stratified* permutation test, shuffling individuals' phenotypes separately within strata of similarly genotyped individuals.

A common question concerns the appropriate number of permutation replicates. A larger number of permutation replicates gives greater precision in the estimated significance thresholds or empirical p -values. While we generally use 1000 permutation replicates initially, we may go up to 10,000 or even 100,000 replicates in order to achieve greater precision. Note that the number

of permutation replicates that meet or exceed a given LOD score follows a $\text{binomial}(r, p)$ distribution, with r being the number of replicates and p being the true p -value. In the case that the p -value is around 0.05, the standard error of the estimated p -value from the permutation test will be around $\sqrt{0.05 \times 0.95/r}$. Thus, to achieve a standard error of 0.005, one would need $0.05 \times 0.95/(0.005)^2 = 1900$ permutation replicates.

Finally, we wish to emphasize that we strongly oppose the use of strict thresholds for statistical significance; it is better to report genome-scan-adjusted p -values. P -values of 4.6% and 5.4% are essentially the same and so should be treated the same. They shouldn't be called "significant" and "suggestive" according to whether they landed below or above a 5% cutoff. Moreover, the importance accorded to a particular p -value depends upon a number of factors, including the ultimate goals of the experiment.

Example

Let us illustrate the permutation test by considering the `hyper` data. We can perform a permutation test with any of the QTL mapping methods discussed in this chapter through the argument `n.perm` to the `scanone` function. That is, we may use the `scanone` function just as before, but by including `n.perm=1000` in the code, the function performs a permutation test with 1000 replicates rather than doing the genome scan with the data. We must, of course, have previously run `calc.genoprob` to obtain the QTL genotype probabilities; let's reload the `hyper` data and run `calc.genoprob` again, just to be sure.

```
> data(hyper)
> hyper <- calc.genoprob(hyper, step=1, error.prob=0.001)
```

Now we use `scanone` to do the permutation test. We use `verbose=FALSE` to suppress any output.

```
> operm <- scanone(hyper, n.perm=1000, verbose=FALSE)
```

The result is a vector of length 1000, containing the genome-wide maximum LOD score from each of 1000 permutation replicates. We may look at the first 5 results as follows.

```
> operm[1:5]
[1] 1.547 1.434 4.184 2.151 1.060
```

The object `operm` has class "`scanoneperm`". There is a corresponding `plot` function, for plotting a histogram of the results, and a `summary` function, for calculating LOD thresholds.

A histogram of the permutation results is produced as follows; see Fig. 4.21.

```
> plot(operm)
```

To obtain estimated genome-wide LOD thresholds for significance levels 20% and 5%, we do the following.

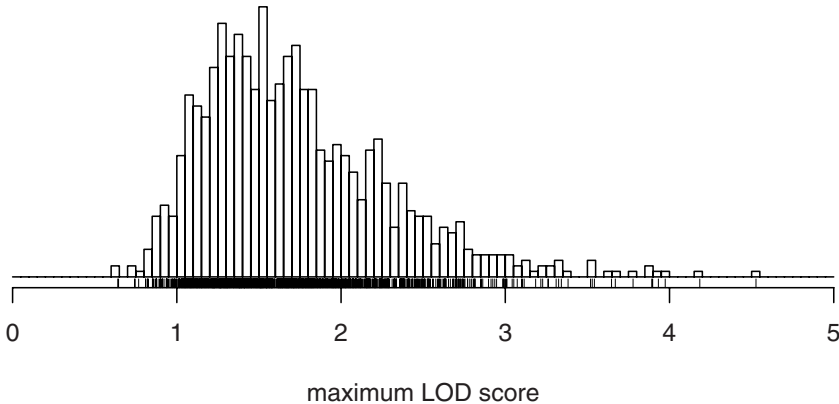


Figure 4.21. Histogram of the genome-wide maximum LOD scores from 1000 permutation replicates with the `hyper` data. The LOD scores were calculated by Haley–Knott regression.

```
> summary(operm, alpha=c(0.20, 0.05))
```

```
LOD thresholds (1000 permutations)
```

```
lod
```

```
20% 2.16
```

```
5% 2.76
```

In the above, we used the traditional permutation test. However, for the `hyper` data, a selective genotyping strategy was used, and so it is best to use a stratified permutation test, permuting individuals' phenotypes separately within strata defined by the extent of genotyping.

We must first define a vector that indicates the strata. This may be done as follows. We place individuals who were genotyped at more than 100 markers in one group and the other individuals in a second group.

```
> strat <- (ntyped(hyper) > 100)
```

We then rerun the permutation test using the argument `perm.strata` to indicate the strata.

```
> operms <- scanone(hyper, n.perm=1000, perm.strata=strat,
+                   verbose=FALSE)
```

In this particular case, we see little difference in the significance threshold when using the stratified permutation test. The 5% threshold is 2.71 (versus 2.76 from the traditional permutation test).

```
> summary(operms, alpha=c(0.20, 0.05))
```

LOD thresholds (1000 permutations)

```
lod
20% 2.10
5% 2.71
```

Turning now to the use of the permutation results, note that if we provide these results to the `summary.scanone` function, we can have LOD thresholds calculated automatically. For example, the following picks out the LOD peaks (no more than one per chromosome) that meet the 10% significance level.

```
> summary(out.em, perms=operms, alpha=0.1)
```

```
chr pos lod
c1.loc45 1 48.3 3.53
D4Mit164 4 29.5 8.09
```

We may further obtain the genome-scan-adjusted p -value for each LOD peak.

```
> summary(out.em, perms=operms, alpha=0.1, pvalues=TRUE)
```

```
chr pos lod pval
c1.loc45 1 48.3 3.53 0.008
D4Mit164 4 29.5 8.09 0.000
```

For the QTL on chromosome 4, our estimated p -value is 0, as no permutations showed a LOD score of 8.1 or greater. Citing a p -value of 0 doesn't seem right, but we can get an upper confidence limit on the true p -value with the `binom.test` function, as follows.

```
> binom.test(0, 1000)$conf.int
```

```
[1] 0.000000 0.003682
```

Thus, we might report $p < 0.004$.

4.4 The X chromosome

The X chromosome exhibits special behavior and must be treated differently from the autosomes in QTL mapping. The behavior of the X chromosome depends on the direction of the cross, as well as the sex of the progeny. We enumerate the possibilities in Fig. 4.22 and 4.23.

In Fig. 4.22, the four possible backcrosses to a single strain are presented. For the backcrosses in panels c and d, in which the F_1 parent is male, the X chromosome is not subject to recombination, and so one cannot map QTL on the X chromosome. We omit these crosses from further consideration. In panels a and b, in which the F_1 parent is female, the X chromosome does recombine, and the order of the cross producing the F_1 parent has no impact

on the behavior of the X chromosome. Note that the backcross males are hemizygous A or B, while females have genotype AA or AB. Thus, rather than comparing, as for the autosomes, the phenotypic means between the AA and AB genotype groups, the X chromosome requires a comparison of the phenotypic means across four genotypic groups.

In Fig. 4.23, the four possible intercrosses are presented. In all cases, F_2 progeny have a single X chromosome subject to recombination, and male F_2 progeny are, at any given locus, hemizygous A or B. In the case that the F_1 male parent was derived from a cross $A \times B$ (with the A parent being female; panels a and b), the female F_2 progeny are either AA or AB. In the cases that the F_1 male was derived from a cross $B \times A$ (with the B parent being female; panels c and d), the female F_2 progeny are either BB or AB. Note that the direction of the cross giving the female F_1 parent does not affect the behavior of the X chromosome in the F_2 progeny. Thus, when we discuss the direction of the intercross, we consider only the direction of the cross that produced the male F_1 parent. The crosses in Fig. 4.23, panels a and b, are treated the same, and the crosses in Fig. 4.23, panels c and d, are treated the same.

The relevant genotype comparisons are somewhat different for the X chromosome than for autosomes. Further, the null hypothesis of no linkage must be reformulated to avoid spurious linkage to the X chromosome as a result of sex- or cross-direction-differences in the phenotype. (Sex differences are observed in many phenotypes, and systematic phenotypic differences between reciprocal crosses may arise, for example, from parent-of-origin effects. If not taken into account, such systematic differences can lead to large LOD scores on the X chromosome even in the absence of X chromosome linkage.) Finally, to account for the fact that the number of degrees of freedom for the linkage test on the X chromosome may be different from that on the autosomes, an X-chromosome-specific significance threshold is required.

4.4.1 Analysis

The choice of the null and alternative hypotheses requires careful thought. The goal of the R/qtl implementation, which we describe here, is to have a procedure for routine use in QTL mapping. The choice of genotype comparisons was based on the following basic principles. First, a sex- or cross-direction-difference in the phenotype should not lead to spurious linkage to the X chromosome. Second, the set of comparisons should be parsimonious but reasonable. Third, the null hypothesis must be nested within the alternative hypothesis. The choices for all possible cases are presented in Table 4.3; we explain how these choices were made through specific examples, below.

Consider, for example, an intercross performed in one direction and including both sexes. Females then have X chromosome genotype AA or AB, while males are hemizygous A or B. Males that are hemizygous B should be treated separately from the AA or AB females, and so the null hypothesis must then allow for a sex-difference in the phenotype, as otherwise, the presence of such

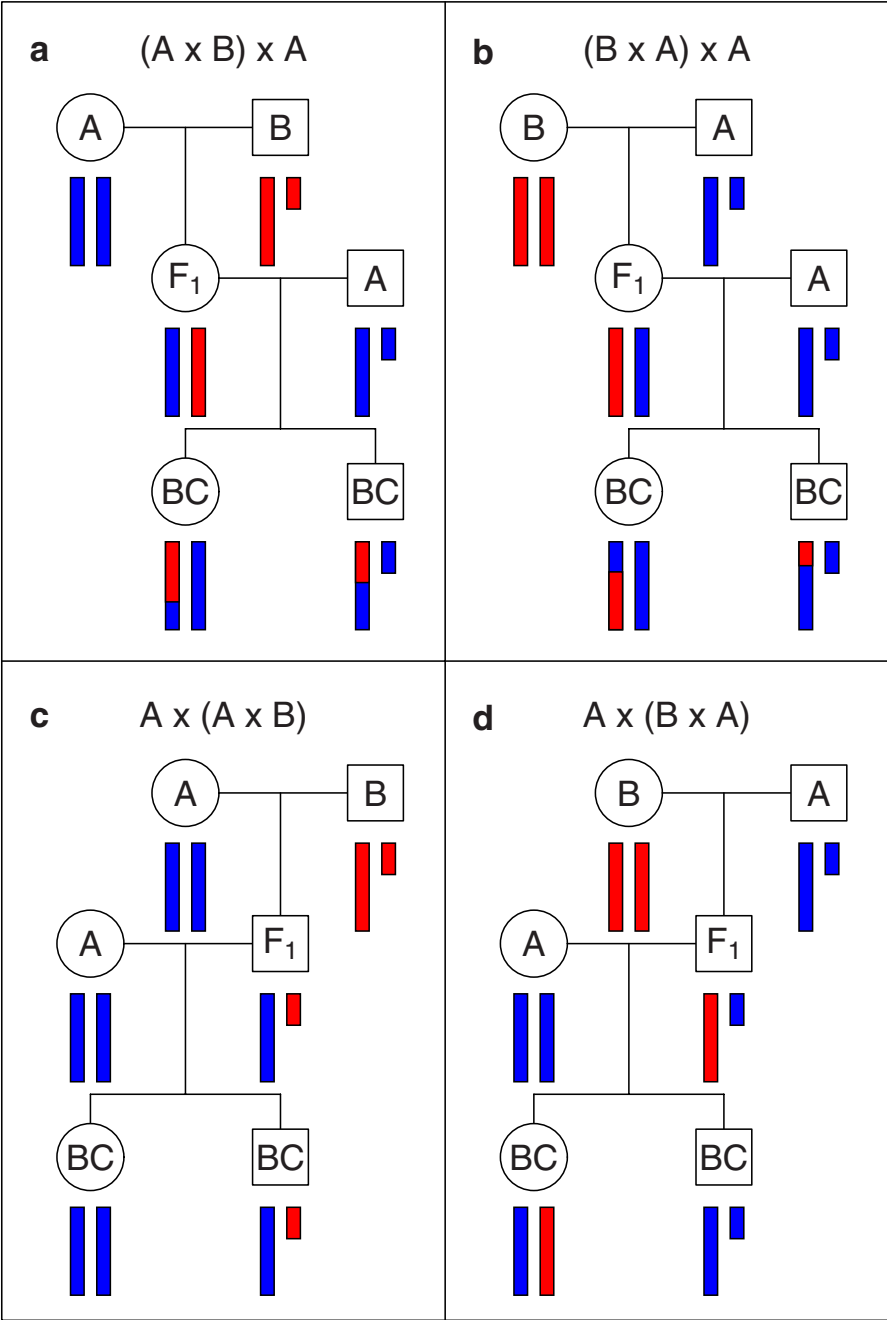


Figure 4.22. The behavior of the X chromosome in a backcross. Circles and squares correspond to females and males, respectively. Blue and red bars correspond to DNA from strains A and B, respectively. The small bar is the Y chromosome.

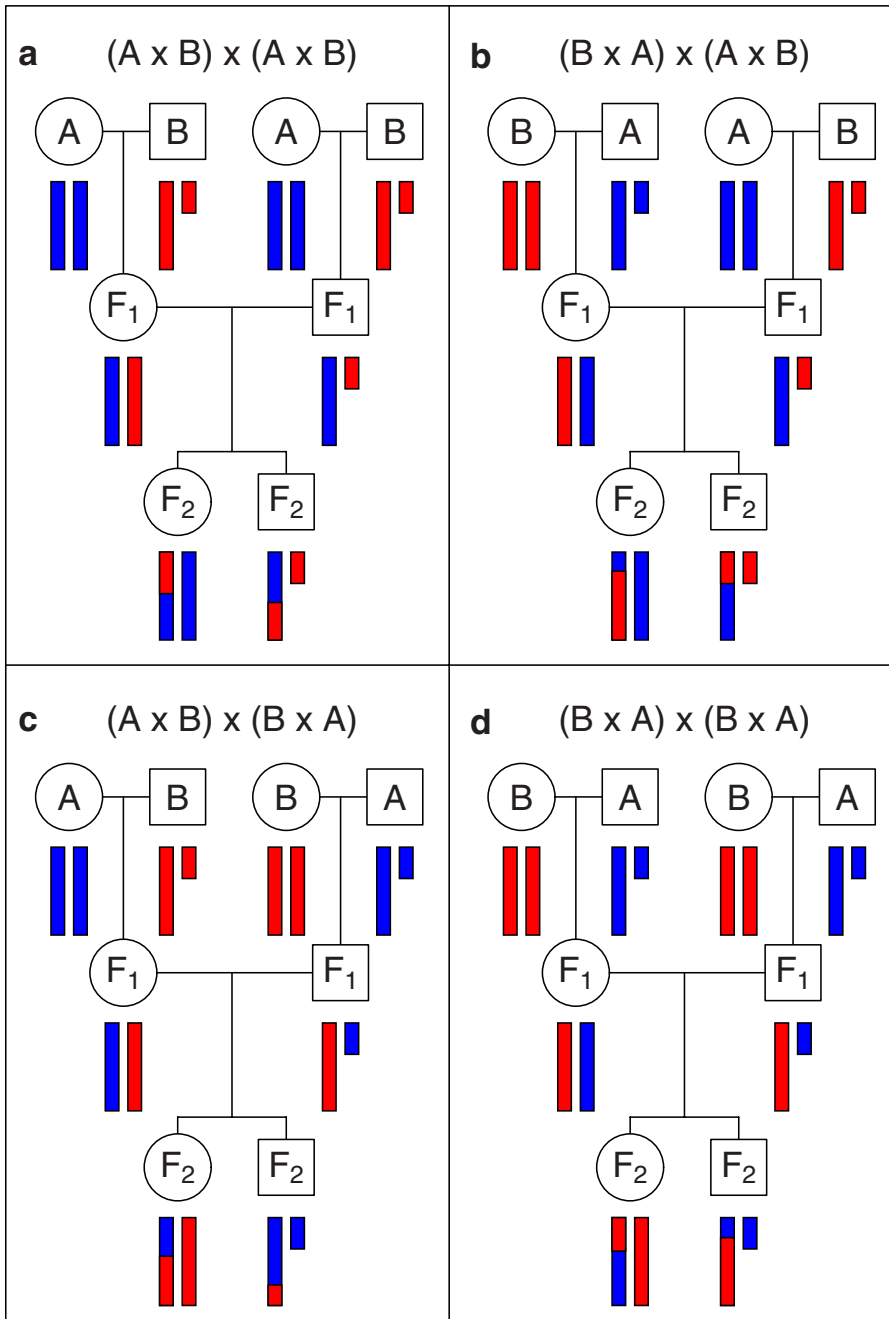


Figure 4.23. The behavior of the X chromosome in an intercross. Circles and squares correspond to females and males, respectively. Blue and red bars correspond to DNA from strains A and B, respectively. The small bar is the Y chromosome.

Table 4.3. Contrasts for analysis of the X chromosome in standard crosses.

Cross	Direction	Sexes	Contrasts	H ₀	df
BC		Both	AA:AB:AY:BY	$\bar{\varphi}:\bar{\sigma}$	2
BC		$\bar{\varphi}$	AA:AB	Grand mean	1
BC		$\bar{\sigma}$	AY:BY	Grand mean	1
F ₂	Both	Both	AA:ABf:ABr:BB:AY:BY	$\bar{\varphi}_f:\bar{\varphi}_r:\bar{\sigma}$	3
F ₂	Both	$\bar{\varphi}$	AA:ABf:ABr:BB	$\bar{\varphi}_f:\bar{\varphi}_r$	2
F ₂	Both	$\bar{\sigma}$	AY:BY	Grand mean	1
F ₂	One	Both	AA:AB:AY:BY	$\bar{\varphi}:\bar{\sigma}$	2
F ₂	One	$\bar{\varphi}$	AA:AB	Grand mean	1
F ₂	One	$\bar{\sigma}$	AY:BY	Grand mean	1

a sex difference would cause spurious linkage to the X chromosome. For the null hypothesis to be nested within the alternative, the alternative must allow separate phenotype averages for the genotype groups AA, AB, AY, and BY (see Table 4.3). We will call this the contrast AA:AB:AY:BY. Note that there are two degrees of freedom for this test of linkage, just as for the autosomes, as there are four mean parameters under the alternative and two under the null (the average phenotype for each of females and males).

A somewhat more complex example is for the case that both directions of the intercross were performed, but only females were phenotyped. As the AA individuals come from one direction of the cross (which we will call the “forward” direction) and the BB individuals come from the other direction of the cross (the “reverse” direction), a cross direction effect on the phenotype would cause spurious linkage to the X chromosome if the null hypothesis did not allow for that effect. But then, in order for the null hypothesis to be nested within the alternative, the AB individuals from the two cross directions must be allowed to be different. Thus we arrive at the contrasts AA:ABf:ABr:BB for the alternative and forward:reverse for the null. Here, again, the linkage test has two degrees of freedom.

In the analogous case with males only, since both directions give rise to equal parts hemizygous A and hemizygous B individuals, we need not split the individuals according to the direction of the cross, as a cross direction effect cannot cause spurious linkage to the X chromosome. In this case, the test for linkage has one degree of freedom.

In the most complex case, of an intercross with both directions and both sexes, all four types of females must be allowed to be separate, whereas the males from the two directions may be pooled, and so the simplest comparison includes the contrasts AA:ABf:ABr:BB:AY:BY, with the null hypothesis using the contrasts female forward:female reverse:male. Thus, the linkage test has three degrees of freedom.

For all interval mapping methods, the actual analysis is essentially the same for the X chromosome as for the autosomes. As each backcross or intercross individual has a single X chromosome that was subject to recombination,

the calculation of the genotype probabilities given the available multipoint marker genotype data is identical to those for an autosome in a backcross, and so nothing new is needed there. The further analysis is hardly modified; the only changes concern the set of genotype groups and the possible inclusion of sex and/or cross direction as covariates under the null hypothesis. Provided that phenotypes **sex** and **pgm** (if necessary) are included with the data (see Sec. 2.1.1, page 24), the analysis will be performed correctly without any intervention from the user.

4.4.2 Significance thresholds

A further point concerns the need for X-chromosome-specific levels of significance, as the number of degrees of freedom for the X chromosome can differ from that for the autosomes. We can assign a chromosome-specific false positive rate of α_i for chromosome i . We require, however, that the α_i are chosen in order to maintain the desired genome-wide significance level, α . Under the null hypothesis of no QTL and with the assumption of independent assortment of chromosomes, the LOD scores on separate chromosomes are independent, and so we must choose the α_i so that $\alpha = 1 - \prod_i (1 - \alpha_i)$.

Any choice of the α_i satisfying this equation will provide a genome-wide false positive rate that is maintained at the desired level. For example, one could choose $\alpha_1 = \alpha$ and $\alpha_i = 0$ for $i \neq 1$. A key issue, in choosing the α_i , concerns the power to detect a QTL. In the preceding example, one would have high power to detect a QTL on chromosome 1, but no power to detect a QTL on any other chromosome. The usual approach, with a constant LOD threshold across the genome, provides high power to detect a QTL irrespective of its location: in the case of high and uniform marker density and the presence of a single autosomal QTL, the power to detect the QTL would be the same no matter where it resides.

A reasonable approach is to use $\alpha_i = 1 - (1 - \alpha)^{L_i/L}$, where L_i is the genetic length of chromosome i and $L = \sum_i L_i$. This corresponds approximately to the use of a constant LOD threshold across the autosomes.

Our actual recommendation is slightly different: we use a constant LOD threshold across the autosomes and a separate threshold for the X chromosome. Taking L_A to be the sum of the genetic lengths of the autosomes and L_X to be the length of the X chromosome, so that $L = L_A + L_X$, we use $\alpha_A = 1 - (1 - \alpha)^{L_A/L}$ and $\alpha_X = 1 - (1 - \alpha)^{L_X/L}$. In particular, in a permutation test to determine LOD thresholds, one would calculate, for permutation replicate j , LOD_{jA}^* as the maximum LOD score across all autosomes and LOD_{jX}^* as the maximum LOD score across the X chromosome. The LOD threshold for the autosomes would be the $1 - \alpha_A$ quantile of the LOD_{jA}^* , and the LOD threshold for the X chromosome would be the $1 - \alpha_X$ quantile of the LOD_{jX}^* .

Genome-scan-adjusted p -values can be estimated from the permutation results as follows. For a putative QTL on an autosome, one would first calculate

the proportion, call it p , of the LOD_{jA}^* that were greater or equal to the observed LOD score. The adjusted p -value would then be $1 - (1 - p)^{L/L_A}$. Equations for a locus on the X-chromosome are analogous, replacing the A 's with X 's.

The precise estimation of the X-chromosome-specific LOD threshold will require considerably more permutation replicates. We have found that one must use roughly L/L_X times more permutation replicates to get the same precision for an adjusted p -value for the X chromosome as one would typically need if a constant LOD threshold were used across the genome.

We have neglected to mention an important detail in the permutations concerning the X chromosome. The rows in the phenotype matrix are shuffled relative to the genotype data. Thus the `sex` (and `pgm`) attached to a particular phenotype is preserved. The X chromosome genotypes, however, are randomized between males and females, but in a special way: the X chromosome genotypes are coded as 1/2 for all individuals, indicating the pattern of recombination and of missing genotype data. These are shuffled across all individuals, but the 1/2 codes are still interpreted as AA/AB for females from one direction of the cross, BB/AB for females from the other direction, and AY/BY for males.

An alternate strategy would be to permute separately within the four strata (males and females in each cross direction). This would be important if there were differences in the pattern of genotype data among the strata.

4.4.3 Example

As an example, we consider the data of Grant *et al.* (2006), which concerns the basal iron levels in the liver and spleen of intercross mice. Both sexes from reciprocal intercrosses with the C57BL/6J/Ola and SWR/Ola strains were used; there are 284 individuals in total. The data are available in the R/qtlbook package as the `iron` data. There are two phenotypes: the level of iron (in $\mu\text{g/g}$) in the liver and spleen. There are approximately equal proportions of males and females and of mice from each cross direction.

We can get access to the data and make a summary plot as follows; see Fig. 4.24. We use `pheno=1:2` to show histograms of just the liver and spleen phenotypes, suppressing barplots of `sex` and `pgm`.

```
> library(qtlbook)
> data(iron)
> plot(iron, pheno=1:2)
```

Figure 4.25 contains a scatterplot of the $\log_2(\text{liver})$ and $\log_2(\text{spleen})$ phenotypes, with females as red circles and males as blue $\times$'s. The figure was obtained as follows.

```
> plot(log2(liver) ~ log2(spleen), data=iron$pheno,
+      col=c("red", "blue")[iron$pheno$sex],
+      pch=c(1,4)[iron$pheno$sex])
```

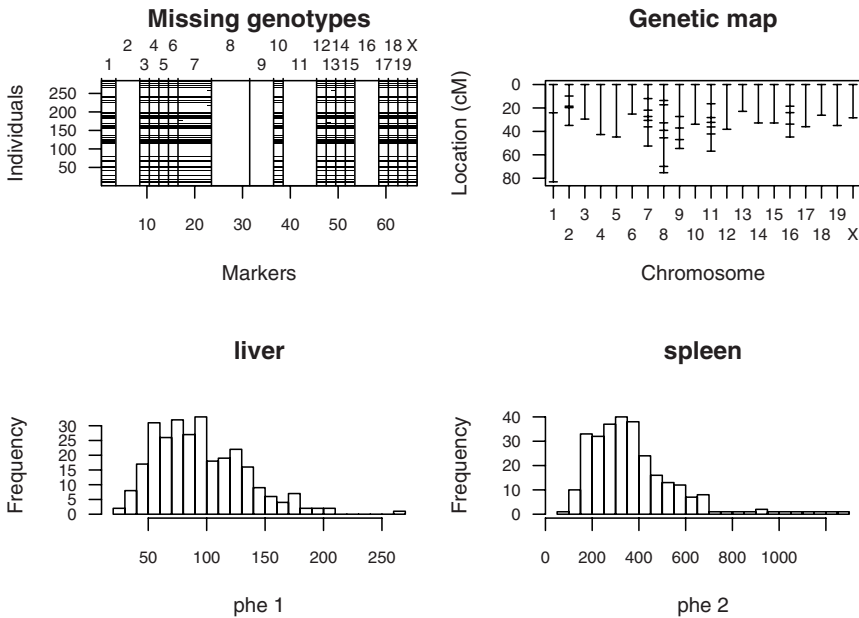


Figure 4.24. The summary plot of the *iron* data.

We use `col` and `pch` to indicate the color and plotting characters to use. For example, `c(1,4)[iron$pheno$sex]` gives a vector of 1's and 4's according to the sexes of the individuals, and with `pch`, 1 corresponds to a circle and 4 to an $\times$.

Note that both phenotypes show a large sex difference, with females having larger iron levels than males. For example, the average iron levels in the liver of females and males are 112 (SE = 3) and 78 (SE = 3), respectively. If the sex differences were not taken into account in QTL mapping on the X chromosome, the LOD scores for that chromosome would be increased by 12.9 and 4.9 for the liver and spleen phenotypes, respectively.

We will focus on the liver phenotype, but we will consider it on the $\log_2$ scale. (Discussion of the analysis of the spleen phenotype is deferred to Sec. 4.7.) We transform the phenotype, and then use `calc.genoprob` and `scanone` to perform a genome scan by standard interval mapping.

```
> iron$pheno[,1] <- log2(iron$pheno[,1])
> iron <- calc.genoprob(iron, step=1, error.prob=0.001)
> out.liver <- scanone(iron)
```

A plot of the results, obtained with `plot.scanone`, is shown in Fig. 4.26. The peaks with LOD > 3 may be obtained with `summary.scanone`.

```
> summary(out.liver, 3)
```

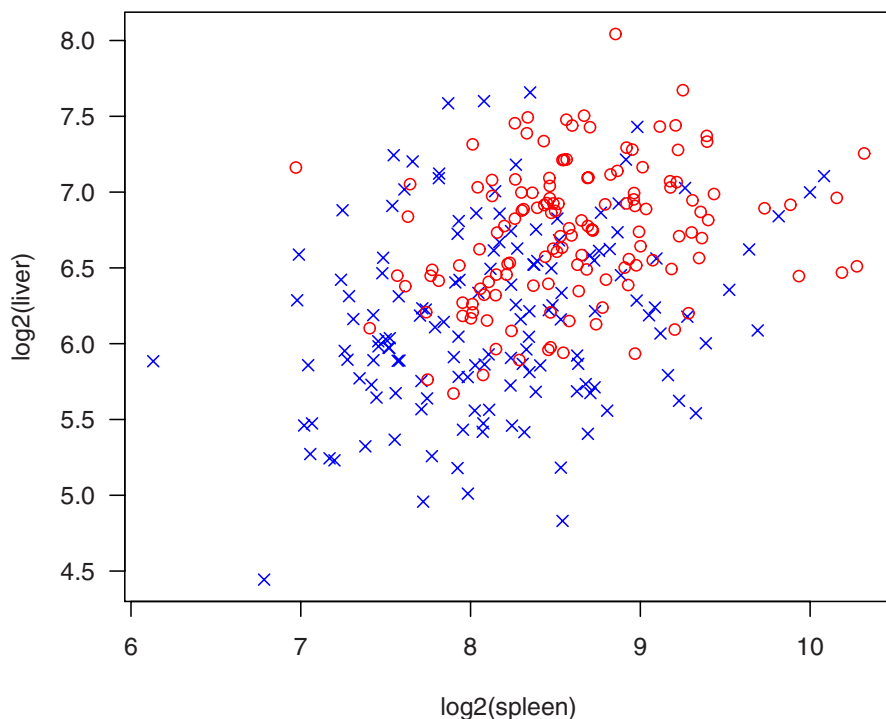


Figure 4.25. Scatterplot of $\log_2(\text{liver})$ versus $\log_2(\text{spleen})$ for the `iron` data, with females as red circles and males as blue $\times$'s.

	chr	pos	lod
D2Mit17	2	56.8	5.09
c7.loc47	7	48.1	3.41
D8Mit294	8	39.1	3.27
c16.loc22	16	28.6	9.47

The maximum LOD score on the X chromosome (0.31) is not so interesting. Nevertheless, it is valuable to show how the permutation test would be done to get autosome- and X-chromosome-specific LOD thresholds. We again use the `scanone` function and use the `n.perm` argument to indicate the number of permutations to perform, but here we also use `perm.Xsp=TRUE` to indicate that we want to perform X-chromosome-specific permutations. In this case, `n.perm` indicates the number of permutations to perform for the autosomes. For the X chromosome, `n.perm  $\times$   $L_A/L_X$` permutations are performed.

```
> operm.liver <- scanone(iron, n.perm=1000, perm.Xsp=TRUE,
+                           verbose=FALSE)
```

The LOD thresholds for a 5% significance level may be obtained as follows.

```
> summary(operm.liver, alpha=0.05)
```

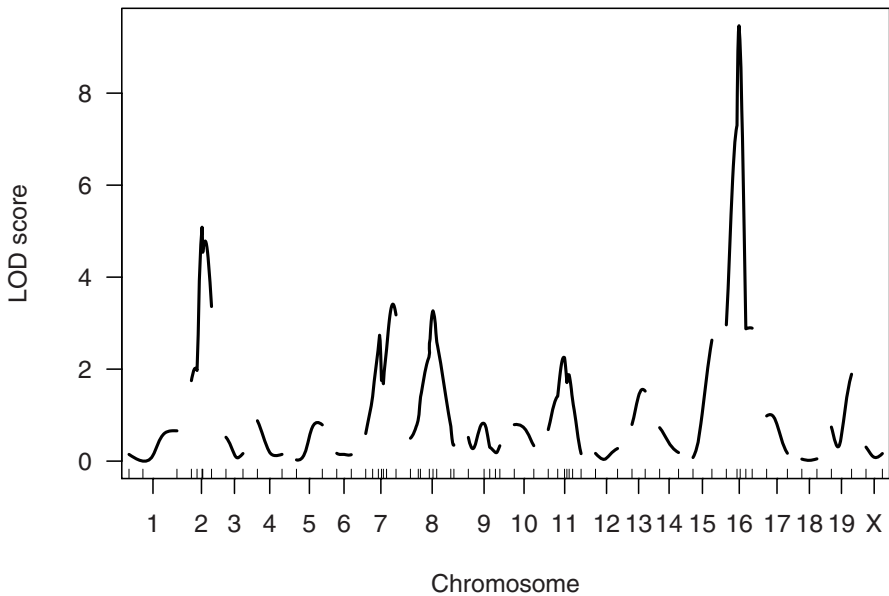


Figure 4.26. LOD curves for the liver phenotype in the `iron` data, calculated by standard interval mapping.

Autosome LOD thresholds (1000 permutations)

```
lod
5% 3.36
```

X chromosome LOD thresholds (28243 permutations)

```
lod
5% 4.83
```

The threshold for the X chromosome is much larger than that for the autosomes, as the test for linkage on the X chromosome has three degrees of freedom, while for the autosomes there are two degrees of freedom (see Table 4.3 on page 112).

We can again use the permutation results with `summary.scanone` to automatically calculate the relevant LOD thresholds corresponding to a particular significance level and to obtain genome-scan-adjusted *p*-values.

```
> summary(out.liver, perms=operm.liver, alpha=0.05,
+         pvalues=TRUE)
```

	chr	pos	lod	pval
D2Mit17	2	56.8	5.09	0.00311
c7.loc47	7	48.1	3.41	0.04449
c16.loc22	16	28.6	9.47	0.00000

The alternate strategy, mentioned above, of a stratified permutation test, is performed by first creating a numeric vector that indicates the strata to which the individuals belong.

```
> strat <- as.numeric(iron$pheno$sex) + iron$pheno$pgm*2
> table(strat)
```

```
strat
 1  2  3  4
74 75 65 70
```

We then rerun the permutation test with `scanone`, indicating the strata via the `perm.strata` argument.

```
> operm.liver.strat <- scanone(iron, n.perm=1000, perm.Xsp=TRUE,
+                             perm.strata=strat, verbose=FALSE)
```

The LOD thresholds are not too different.

```
> summary(operm.liver.strat, alpha=0.05)
```

Autosome LOD thresholds (1000 permutations)

```
lod
5% 3.41
```

X chromosome LOD thresholds (28243 permutations)

```
lod
5% 4.51
```

4.5 Interval estimates of QTL location

Once one has obtained evidence for a QTL, one may seek an interval estimate of the location of the QTL. The estimated QTL location by interval mapping will deviate somewhat from the true location, and so we seek the range of possible QTL locations that are supported by the data. We assume, in this section, that there is one and only one QTL on the chromosome of interest.

There are two major methods for calculating an interval estimate of QTL location: LOD support intervals and Bayes credible intervals. The 1.5-LOD support interval is the interval in which the LOD score is within 1.5 units of its maximum. See Fig. 4.27 for an illustration. If the LOD score drops down and back up (as in Fig. 4.27), one would obtain a set of disjoint intervals, but we use the conservative approach of taking the wider connected interval. The amount to drop affects the coverage of the LOD support intervals; we prefer to use 1.5-LOD support intervals for a backcross, and 1.8-LOD support intervals for an intercross.

An approximate Bayes credible interval is obtained by viewing 10^{LOD} as a real likelihood function for QTL location. (In fact, it is a profile likelihood,

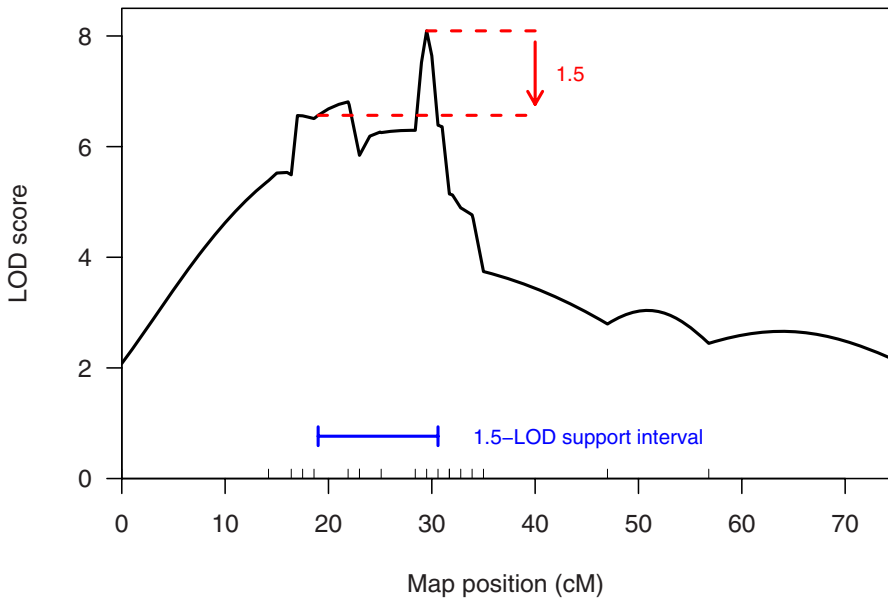


Figure 4.27. Illustration of the 1.5-LOD support interval for chromosome 4 of the *hyper* data. The LOD curve was calculated by standard interval mapping.

as at each point we maximized over the possible QTL effects and the residual SD.) Assuming *a priori* that the QTL is equally likely to be anywhere on the chromosome, the posterior distribution of QTL location is obtained by rescaling 10^{LOD} to be a distribution, $f(\theta \mid \text{data}) = 10^{\text{LOD}(\theta)} / \sum_{\theta} 10^{\text{LOD}(\theta)}$. The 95% Bayes credible interval is defined as the interval I for which $f(\theta \mid \text{data})$ exceeds some threshold and for which $\sum_{\theta \in I} f(\theta \mid \text{data}) \geq 0.95$.

The Bayes interval is illustrated in Fig. 4.28. Plotted is 10^{LOD} , rescaled so that the area underneath the curve is 1. The area shaded in red is approximately 95% of the total area.

It is important to point out that neither the LOD support interval nor the Bayes credible interval behave as true confidence intervals, as interval coverage (the chance of obtaining an interval containing the true QTL location) is not constant, but depends to some extent on the type of cross, marker density, and the size of the QTL effect. Experience has shown, however, that Bayes intervals have remarkably consistent coverage, and so may be preferred.

The use of a nonparametric bootstrap has also been used to create confidence intervals for QTL location. In the nonparametric bootstrap, one samples, with replacement, from the individuals in the cross to create a new data set of the same size as the original, but with some individuals repeated and some omitted. With these new data, interval mapping is performed and the estimated location of the QTL recorded. The process is repeated multiple times, to create a set of QTL location estimates, $\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_b^*$. A confidence interval

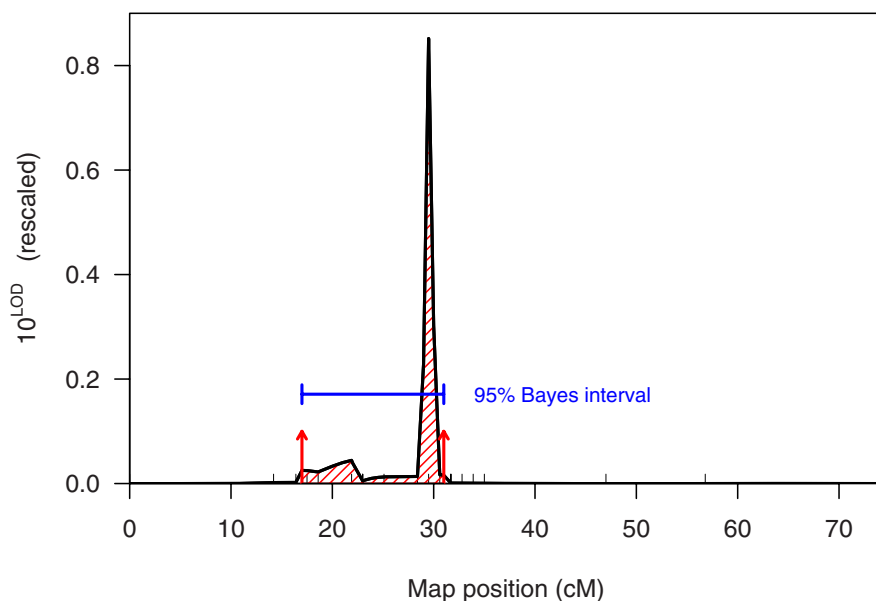


Figure 4.28. Illustration of the approximate 95% Bayes credible interval for chromosome 4 of the **hyper** data. The LOD curve was calculated by standard interval mapping.

is estimated as the region containing 95% of these bootstrap estimates; say the 2.5th to 97.5th percentiles of the $\hat{\theta}_i^*$.

Unfortunately, the bootstrap has been shown to perform poorly in this context, and so it is not recommended. The maximum likelihood estimate of QTL location obtained from interval mapping has a tendency to occur at a marker. Moreover, the standard error of the estimated location depends on the location of the QTL relative to the typed markers. As a result, the coverage of bootstrap confidence intervals for QTL location depends critically on the location of the QTL relative to the markers. In addition, the bootstrap confidence intervals tend to be much wider than the LOD support and Bayes credible intervals.

Example

We can calculate the LOD support interval and Bayes credible interval with the functions `lodint` and `bayesint`, respectively. These take, as input, results from `scanone` and the chromosome to consider, plus the argument `drop` in `lodint`, indicating the amount to drop in LOD (1.5 by default), and `prob` in `bayesint`, indicating the nominal Bayes fraction (95% by default).

We calculate the 1.5-LOD support and 95% Bayes credible intervals for chromosome 4 in the **hyper** data as follows.

```
> lodint(out.em, 4, 1.5)

      chr  pos  lod
c4.loc19   4 19.0 6.566
D4Mit164   4 29.5 8.092
D4Mit178   4 30.6 6.387

> bayesint(out.em, 4, 0.95)

      chr  pos  lod
c4.loc17   4 17.0 6.562
D4Mit164   4 29.5 8.092
c4.loc31   4 31.0 6.359
```

The first and last rows in the results indicate the ends of the intervals; the middle row is the maximum likelihood estimate of QTL location.

Note that the ends of the intervals generally lie between marker locations. One may use the argument `expandtomarkers` to expand the intervals to the nearest flanking markers, as follows.

```
> lodint(out.em, 4, 1.5, expandtomarkers=TRUE)

      chr  pos  lod
D4Mit286   4 18.6 6.507
D4Mit164   4 29.5 8.092
D4Mit178   4 30.6 6.387

> bayesint(out.em, 4, 0.95, expandtomarkers=TRUE)

      chr  pos  lod
D4Mit108   4 16.4 5.490
D4Mit164   4 29.5 8.092
D4Mit80    4 31.7 5.145
```

We can obtain a bootstrap-based confidence interval for the location of a QTL with the function `scanoneboot`, which takes the same set of arguments as `scanone`, though only a single chromosome (indicated with the argument `chr`) is considered, and the number of bootstrap replicates is indicated with the argument `n.boot`. For example, we perform the bootstrap with chromosome 4 of the `hyper` data as follows.

```
> out.boot <- scanoneboot(hyper, chr=4, n.boot=1000)
```

A histogram of the bootstrap results is shown in Fig. 4.29. Note that 80% of the bootstrap locations coincide with genetic markers. The results provide a poor measure of the uncertainty in QTL location.

The output of `scanoneboot` has class `"scanoneboot"`; the actual confidence interval is obtained with the function `summary.scanoneboot`, as follows.

```
> summary(out.boot)
```

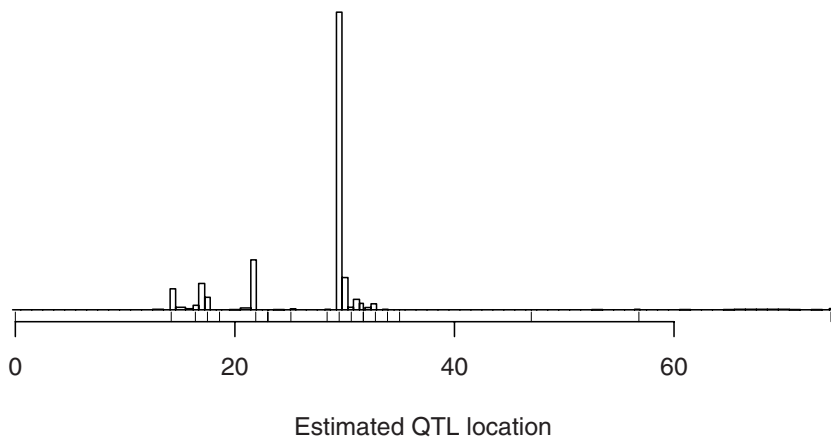


Figure 4.29. Histogram of the estimated QTL locations in 1000 bootstrap replicates with the `hyper` data, chromosome 4, with LOD scores calculated by standard interval mapping. The tick marks beneath the histogram indicate the locations of the genetic markers.

	chr	pos	lod
D4Mit41	4	14.2	6.076
D4Mit164	4	29.5	9.151
D4Mit276	4	32.8	3.881

This is a bit wider than the LOD support and Bayes credible intervals.

4.6 QTL effects

The effect of a QTL is characterized by the difference in the phenotype averages among the QTL genotype groups. For a backcross, we simply look at the difference between the average phenotypes for the heterozygotes and the homozygotes, $a = \mu_{AB} - \mu_{AA}$. For an intercross, with three possible genotypes, one traditionally considers the additive effect, $a = (\mu_{BB} - \mu_{AA})/2$, and the dominance effect, $d = \mu_{AB} - (\mu_{AA} + \mu_{BB})/2$.

In addition, the QTL effect is often characterized by the proportion of the phenotypic variance explained by the QTL (also called the heritability due to the QTL). Specifically, we consider

$$h^2 = \text{var}\{E(y|g)\}/\text{var}(y),$$

where y is the phenotype and g is the QTL genotype. For a backcross, we have $h^2 = a^2/(a^2 + 4\sigma^2)$, where σ^2 is the residual variance. For an intercross, the heritability due to the QTL is $h^2 = (2a^2 + d^2)/(2a^2 + d^2 + 4\sigma^2)$.

Knowledge of the QTL effects is especially important in agricultural investigations to develop improved livestock or crops and in studies of evolution. In biomedical research of models of human disease, which is our primary focus, the QTL effects are of lesser importance: while specific genes may carry over from a model organism to humans, the actual alleles are not likely to be the same, and so the QTL effects in the model are not likely to be relevant for humans. Nevertheless, the QTL effects have important influence on one's ability to fine-map the QTL and ultimately identify the causal gene.

In considering the estimated effects of QTL, it is important to recognize that they are often subject to considerable bias. The point is that, the estimated effect of a QTL will vary somewhat from its true effect, but only when the estimated effect is large will the QTL be detected. (We don't estimate the effects of QTL that we have not detected.) Among those experiments in which the QTL is detected, the estimated QTL effect will be, on average, larger than its true effect. This is *selection bias*.

As an illustration, consider Fig. 4.30. We simulated a backcross with 250 individuals, with a single QTL responsible for either 2.5, 5, or 7.5% of the phenotypic variance. The true QTL effect is indicated with a blue vertical line. The distribution of the estimated QTL effect, across 100,000 simulation replicates, is displayed. There is a direct relationship between the estimated percent variance explained by a QTL and the LOD score (see page 77). With a sample size of 250, a LOD score of 3 corresponds to an estimated phenotypic variance explained of 5.4%. Among the cases in which the LOD score is above 3 (the shaded portions of the distributions), the average estimated effect, indicated by the red vertical line, is larger than the true effect.

Note that the selection bias is largest for QTL with small effects (for which we have lower power for detection). QTL with very large effect are always detected, and so the bias in their estimated effects will be minimal.

For a particular inferred QTL with an estimated effect of moderate size, we cannot know whether it is a weak QTL that we were lucky to detect and whose true effect is really rather small, or a QTL of truly large effect that happened to appear to be not so strong in this particular experiment. The estimated effects of QTL will often be overly optimistic, but we cannot be sure about any particular case.

Selection bias in estimated QTL effects has a number of important implications. First, the overall estimated heritability due to a set of identified QTL will almost always be too large and can be markedly so. Second, investigators are often concerned, after repeating a QTL experiment, that an almost completely different set of QTL were identified. One should not conclude, in such a situation, that the unreplicated QTL were false. Instead, it may be that the phenotype is affected by numerous small-effect QTL, for each of which the power of detection is small, and so in any given experiment we identify a random portion of the QTL. Third, in the consideration of a congenic line (in which, for example, the B allele at a QTL is introgressed into the A strain), one may find that the difference between the average phenotypes for the congenic

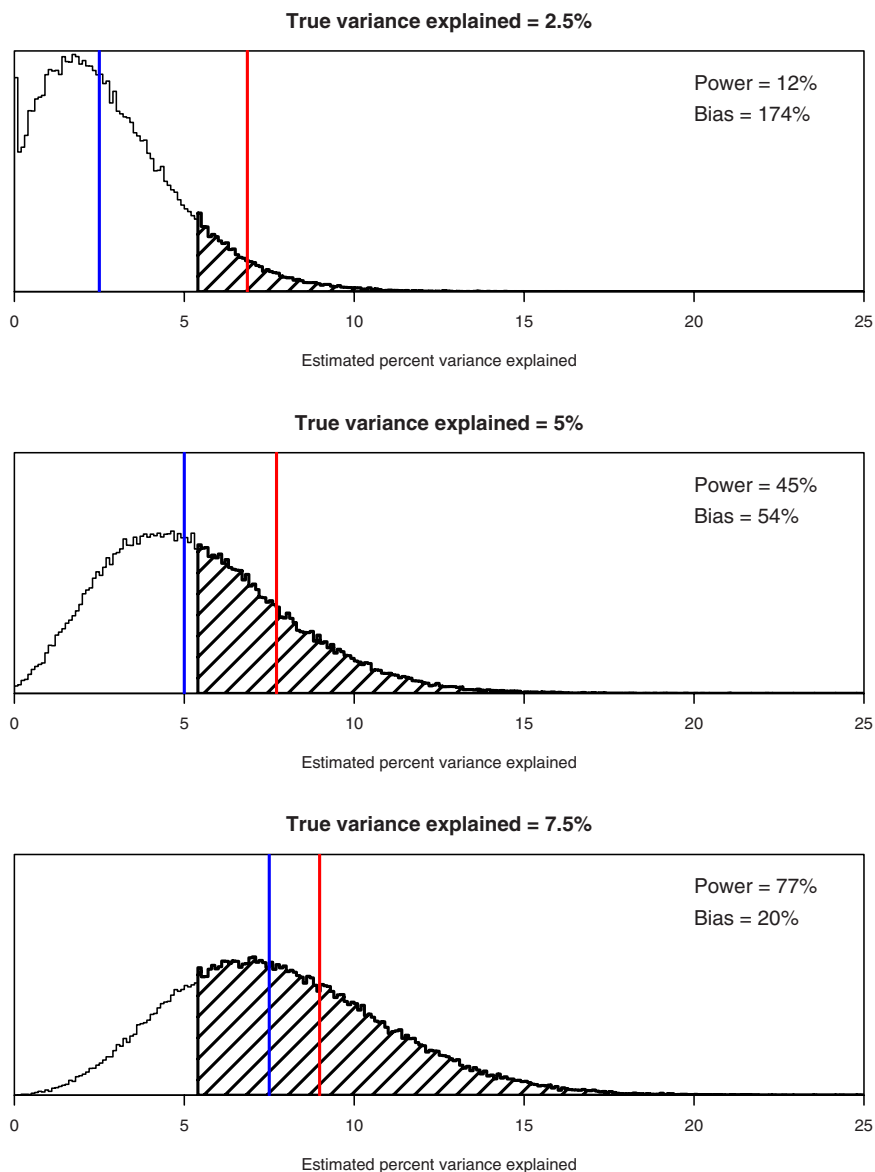


Figure 4.30. Illustration of selection bias in the estimated QTL effect. The curves correspond to the distribution of the estimated percent variance explained by a QTL for different values of the true effect (indicated by the blue vertical lines). The shaded regions correspond to the cases where significant genomewide evidence for the presence of a QTL would be obtained. The red vertical lines indicate the average estimated QTL effect, conditional on the detection of the QTL.

and the recipient strain is not so large as was expected given the QTL mapping results. One might conclude, from this observation, that the QTL consisted of multiple causal genes, and that some have not been included in the congenic line. However, it could just be that the initial estimate of the QTL effect was optimistically large. Finally, in marker-assisted selection efforts, one is likely to find that the progress towards improvement is not so great as had been anticipated, as the true effects of the loci under selection are not so large as their initial estimates.

Example

The function for performing QTL mapping, `scanone`, does not provide estimated QTL effects. Such estimates are best obtained with the function `fitqtl`, particularly for the case of a multiple-QTL model, but we will defer discussion of `fitqtl` to Chap. 9. In the example in this section, we will focus on the use of the function `effectplot`, whose primary purpose is for plotting the phenotype averages for the genotype groups at an inferred QTL.

The function `effectplot` uses the multiple imputation method to obtain estimates of the genotype-specific phenotype averages, taking account of missing genotype data. The estimates are weighted averages of the estimates from the multiple imputations, with the individual imputations being weighted by 10^{LOD} . The standard errors (SEs) include the imputation error. (If imputed genotypes are not available, a call to `sim.geno` with `n.draws=16` is made in order to obtain them.)

The `effectplot` function takes a cross object as input; a marker name is indicated via the `mname1` argument. (The function may be used to plot the estimated effects for two markers, with the second marker indicated via the `mname2` argument; see Chap. 8.) The argument `var.flag` may be used to indicate whether to use a pooled estimate of the residual variance (`var.flag="pooled"`, the default), or to allow different residual variances in the genotype groups (`var.flag="group"`).

One can even use `effectplot` with “pseudomarker” positions (that is, positions on the grid, in between markers). We refer to such pseudomarkers with a chromosome and cM position, in the form “4@29.5”, which refers to the pseudomarker closest to 29.5 cM on chromosome 4.

For the `hyper` data, the largest LOD score was obtained at marker D4Mit164 (chr 4, 29.5 cM). If we had known only the position, we could find the name of the nearest marker with the function `find.marker`, which takes as input a cross object, chromosome, and cM position.

```
> find.marker(hyper, 4, 29.5)
```

```
[1] "D4Mit164"
```

The `effectplot` for marker D4Mit164 in the `hyper` data may be obtained as follows; the result appears in the left panel in Fig. 4.31.

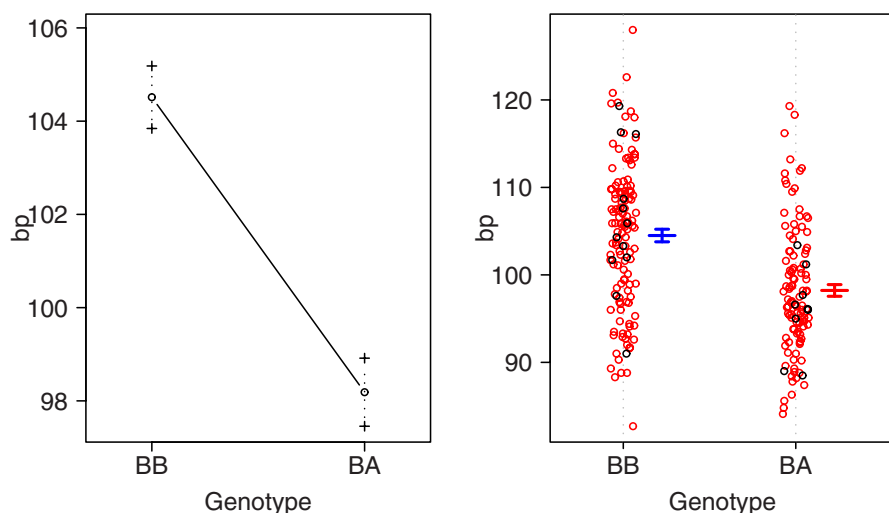


Figure 4.31. Plot of the blood pressure against the genotype at marker D4Mit164 for the `hyper` data. The left panel was produced by `effectplot`. The right panel was produced by `plot.pxd`; red dots correspond to imputed genotypes. Error bars are ± 1 SE.

```
> hyper <- sim.geno(hyper, n.draws=16, error.prob=0.001)
> effectplot(hyper, mname1="D4Mit164")
```

The output of `effectplot` (except for the plot) is generally suppressed, but if one assigns the output to an object, it may be inspected. If one uses `draw=FALSE`, the plot is not created.

```
> eff <- effectplot(hyper, mname1="D4Mit164", draw=FALSE)
> eff
```

```
$Means
D4Mit164.BB D4Mit164.BA
    104.51      98.19
```

```
$SEs
D4Mit164.BB D4Mit164.BA
    0.6703      0.7298
```

The function `plot.pxd` can be used to create a dot plot of a phenotype against the genotypes at a marker. The function is relatively crude in its treatment of missing genotype data: missing genotypes are filled in by a single random imputation, conditional on the available marker data. Thus, if there are many missing genotypes, one should view the results with some skepticism. Genotypes that were imputed are plotted in red.

As input to `plot.pwg`, we again provide a cross object plus a marker name; the marker name is indicated with the argument `marker`. We can plot the phenotype against the genotypes at D4Mit164 as follows. The plot appears in the right panel of Fig. 4.31. Note that most genotypes are missing.

```
> plot.pwg(hyper, marker="D4Mit164")
```

The argument `marker` can take a vector of marker names, in which case the phenotypes will be plotted against the joint genotypes at the markers; see Chap. 8.

4.7 Multiple phenotypes

In a QTL experiment, one often measures multiple related phenotypes. The joint analysis of multiple phenotypes can increase the power for QTL detection and the precision of QTL localization, and can allow one to test for pleiotropy (that a single QTL influences multiple phenotypes) versus tight linkage of distinct QTL. Unfortunately, R/qtl does not yet include facilities for such joint analysis of multiple phenotypes; multiple phenotypes are only considered individually.

By default, `scanone` performs interval mapping with the first phenotype in a data set. A different phenotype may analyzed via the argument `pheno.col` (for “phenotype column”), which is the numeric index of the phenotype to be analyzed or a character string indicating the phenotype by name. One may also use the `pheno.col` argument to select multiple phenotypes for analysis by providing a vector of numeric indices or phenotype names. For most methods, this is accomplished by a loop over the selected phenotypes. However, for Haley–Knott regression (`method="hk"`) and multiple imputation (`method="imp"`), an improvement in computational efficiency is achieved by application of multivariate regression.

In Haley–Knott regression, one regresses the phenotype, y , on the genotype probability matrix, X , by calculating $(X'X)^{-1}X'y$. One may analyze a set of phenotypes, Y , simultaneously, by calculating $(X'X)^{-1}X'Y$. The construction and inversion of $X'X$ need only be done once.

This trick can also be used for permutation tests: one creates a set of permuted phenotypes, pastes them together into one large matrix, and then performs Haley–Knott regression with the permuted phenotypes as a unit. (Thanks are due to Hao Wu who suggested and implemented this strategy.)

Example

To illustrate the analysis of multiple phenotypes, we will return to the `iron` data discussed in Sec. 4.4. There are two phenotypes: the iron levels in the liver and spleen. We may also want to look at these on the log scale; we can place $\log_2(\text{liver})$ and $\log_2(\text{spleen})$ in the phenotypes as follows. Note that `data(iron)` reloads the data.

```
> data(iron)
> iron$pheno <- cbind(iron$pheno[,1:2],
+                    log2liver=log2(iron$pheno$liver),
+                    log2spleen=log2(iron$pheno$spleen),
+                    iron$pheno[,3:4])
```

We place them in positions 3 and 4, moving the `sex` and `pgm` phenotypes to the end.

By default, `scanone` would analyze the first phenotype (liver). If we want to consider the $\log_2(\text{liver})$ phenotype, we would use `pheno.col=3`, as follows.

```
> iron <- calc.genoprob(iron, step=1, error.prob=0.001)
> out.logliver <- scanone(iron, pheno.col=3)
```

We may also refer to phenotypes by name. For example, in place of `pheno.col=3` we could use `pheno.col="log2liver"`, as follows.

```
> out.logliver <- scanone(iron, pheno.col="log2liver")
```

In addition, one may use `pheno.col` with a numeric vector of phenotypes (with length equal to the number of individuals in the cross). And so, if we were interested solely in the analysis of the $\log_2(\text{liver})$ phenotype, we could skip the effort to include the transformed phenotype within the cross object and simply type the following.

```
> out.logliver <- scanone(iron,
+                        pheno.col=log2(iron$pheno$liver))
```

We may analyze all four phenotypes through `pheno.col=1:4`.

```
> out.all <- scanone(iron, pheno.col=1:4)
```

The result has six columns: chromosome, cM position, and then four columns with LOD scores.

```
> out.all[1:5,]
      chr pos liver spleen log2liver log2spleen
D1Mit18  1 27.3 0.1947 0.2695    0.1461    0.034035
c1.loc1  1 28.3 0.1869 0.2602    0.1390    0.026292
c1.loc2  1 29.3 0.1789 0.2515    0.1315    0.019184
c1.loc3  1 30.3 0.1705 0.2436    0.1239    0.012920
c1.loc4  1 31.3 0.1619 0.2366    0.1161    0.007736
```

The `summary.scanone` function displays (by default) the peak positions for the first phenotype, but also shows LOD scores at those positions for the other phenotypes. We can look at the peaks for another phenotype with the `lodcolumn` argument. The LOD columns are indexed as 1, 2, 3, 4, and so to get the peaks above 3 for the $\log_2(\text{spleen})$ phenotype, we would do the following.

```
> summary(out.all, threshold=3, lodcolumn=4)
```

	chr	pos	liver	spleen	log2liver	log2spleen
D8Mit4	8	13.6	0.739	4.3	0.887	3.90
c9.loc50	9	56.6	0.183	10.9	0.199	12.64

Two other summary formats are available, which display the peak LOD scores for all phenotypes. The method mentioned above corresponds to the use of `format="onepheno"`. The use of `format="allpheno"` gives essentially the same output, but with all of the LOD score columns considered. For each LOD score column, we identify the position of the maximum peak and include that row in the output if the LOD score exceeds its threshold. Thus, the output may display multiple rows for a given chromosome. Note that the `threshold` argument may be a single number (applied to all LOD score columns) or a vector specifying separate thresholds for each LOD score column. For example, the following returns the results for positions at which at least one of the LOD score columns achieved its maximum for a chromosome, provided that maximum LOD score exceeded 3.

```
> summary(out.all, threshold=3, format="allpheno")
```

	chr	pos	liver	spleen	log2liver	log2spleen
D2Mit17	2	56.8	4.907	1.917	5.086	2.279
c7.loc47	7	48.1	2.935	0.395	3.413	0.596
D8Mit4	8	13.6	0.739	4.303	0.887	3.895
D8Mit294	8	39.1	3.769	1.902	3.268	1.724
c8.loc40	8	40.0	3.786	1.752	3.241	1.581
c9.loc50	9	56.6	0.183	10.897	0.199	12.642
c16.loc21	16	27.6	7.837	0.848	9.338	1.136
c16.loc22	16	28.6	7.829	0.909	9.465	1.214

Finally, with `format="allpeaks"`, a single row is given for each chromosome, containing the maximum LOD score for each phenotype column and the position at which it was maximized. Those chromosomes for which at least one of the LOD score columns exceeded its threshold are printed. For example, the following returns the chromosomes at which at least one of the LOD score columns had a peak exceeding 3.

```
> summary(out.all, threshold=3, format="allpeaks")
```

	chr	pos	liver	pos	spleen	pos	log2liver	pos	log2spleen
2	2	56.8	4.907	58.0	1.994	56.8	5.086	58.0	2.42
7	7	50.1	2.959	53.6	0.818	48.1	3.413	53.6	1.01
8	8	40.0	3.786	13.6	4.303	39.1	3.268	13.6	3.90
9	9	31.6	0.998	56.6	10.897	32.6	0.824	56.6	12.64
16	16	27.6	7.837	30.6	0.981	28.6	9.465	30.6	1.31

If `scanone` output that contains multiple LOD score columns is sent to the `plot.scanone` function, the default is again to plot just the first one. A different column may be indicated via the argument `lodcolumn`, which (as

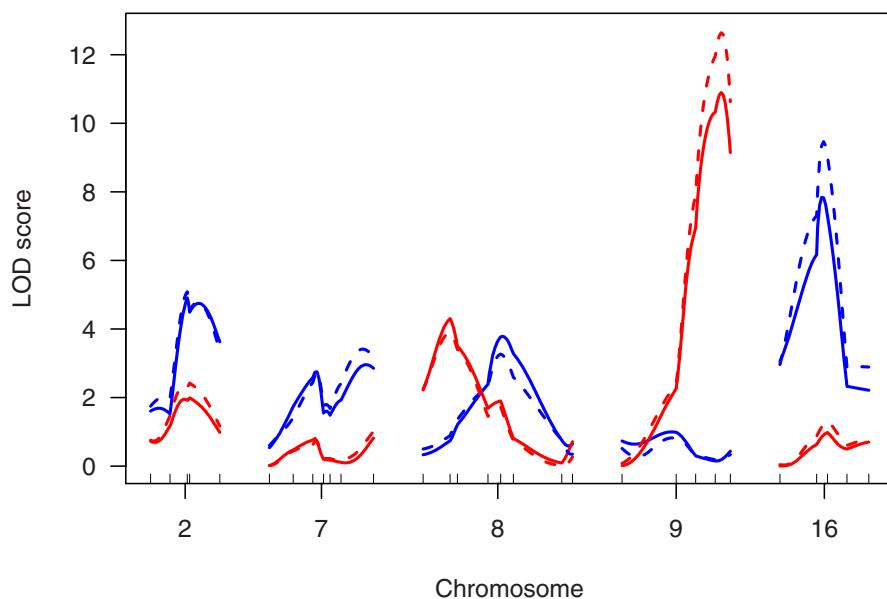


Figure 4.32. LOD curves for selected chromosomes with the `iron` data. Blue and red correspond to the liver and spleen phenotypes, respectively. Solid and dashed curves correspond to the original and log scale, respectively.

in `summary.scanone`) takes values 1, 2, Further, `lodcolumn` can take a vector of up to three values. And so, if we want to look at the results for all four phenotypes, we might do the following. The results are in Fig. 4.32.

```
> plot(out.all, lodcolumn=1:2, col=c("blue", "red"),
+      chr=c(2, 7, 8, 9, 16), ylim=c(0,12.7), ylab="LOD score")
> plot(out.all, lodcolumn=3:4, col=c("blue", "red"), lty=2,
+      chr=c(2, 7, 8, 9, 16), add=TRUE)
```

Note the use of `ylim` to set the y -axis limits, to ensure that all of the LOD curves would be contained in the plot.

We can use `scanone` to perform permutation tests on the set of phenotypes simultaneously. As discussed in Sec. 4.4, we should perform separate permutations on the autosomes and X chromosome, which may be accomplished by using `perm.Xsp=TRUE`.

```
> operm.all <- scanone(iron, pheno.col=1:4, n.perm=1000,
+                      perm.Xsp=TRUE)
```

We may again use `summary` to obtain LOD thresholds for each phenotype for the autosomes and the X chromosome.

```
> summary(operm.all, alpha=0.05)
```

Autosome LOD thresholds (1000 permutations)

	liver	spleen	log2liver	log2spleen
5%	3.96	7.27	3.4	3.37

X chromosome LOD thresholds (28243 permutations)

	liver	spleen	log2liver	log2spleen
5%	3.72	6.57	4.68	4.63

The unusually large LOD thresholds for the spleen phenotype are due to the occurrence of spuriously high LOD scores in regions with large gaps between markers (see Sec. 4.2.5). The log transformed phenotypes are preferred for these data, due to the skewed phenotype distributions (see Fig. 4.24 on page 115).

The permutation results may again be used in `summary.scanone` to automatically calculate thresholds and to obtain genome-scan-adjusted p -values. The significance level for the LOD thresholds is indicated by the argument `alpha`, which may take only one value, applied to all LOD score columns. For example, the following gives, for each chromosome, the maximum LOD score from each LOD score column with the corresponding genome-scan-adjusted p -values. The results for a chromosome are printed if at least one of the LOD score columns exceeded its 5% LOD threshold.

```
> summary(out.all, format="allpeaks", perms=perm.all,
+         alpha=0.05, pvalues=TRUE)
```

	chr	pos	liver	pval	pos	spleen	pval	pos	log2liver
2	2	56.8	4.907	0.0166	58.0	1.994	0.833	56.8	5.086
7	7	50.1	2.959	0.2104	53.6	0.818	1.000	48.1	3.413
8	8	40.0	3.786	0.0641	13.6	4.303	0.320	39.1	3.268
9	9	31.6	0.998	0.9992	56.6	10.897	0.000	32.6	0.824
16	16	27.6	7.837	0.0000	30.6	0.981	0.999	28.6	9.465
		pval	pos	log2spleen	pval				
2		0.00104	58.0		2.42	0.2525			
7		0.04759	53.6		1.01	0.9967			
8		0.06826	13.6		3.90	0.0114			
9		1.00000	56.6		12.64	0.0000			
16		0.00000	30.6		1.31	0.9513			

We see evidence for QTL on chromosomes 2, 7, 8 and 16 for $\log_2(\text{liver})$ and on chromosomes 8 and 9 for $\log_2(\text{spleen})$.

4.8 Summary

In a single-QTL scan, we posit the presence of a single QTL and consider each position, one at a time, as the putative location of that QTL. With

dense markers and complete marker genotype data, one may perform analysis of variance at each marker. More typically, however, markers are spaced at 10–20 cM, and some marker genotypes are missing. There are several approaches to interval mapping, for interrogating positions between markers. These approaches differ in the way in which they take account of missing genotype data at a putative QTL.

Standard interval mapping may be viewed as the gold standard, but it is susceptible to spurious linkage peaks in regions of low marker information when the phenotype distribution is not approximately normal. Haley–Knott regression often gives a good approximation to standard interval mapping, at a great improvement in computation time, but it performs poorly in the case of selective genotyping. The extended Haley–Knott regression method gets around these problems. The multiple imputation approach is rather slow for single-QTL analyses, but will show advantages for the fit and exploration of multiple-QTL models.

Statistical significance in a single-QTL scan is generally established through the consideration of the distribution of the genome-wide maximum LOD score, under the global null hypothesis of no QTL. This distribution is best derived via a permutation test.

There are several technical difficulties that arise in the analysis of the X chromosome. Additional covariates need to be incorporated into the null hypothesis, to avoid spurious linkage to the X chromosome, and a separate significance threshold will generally be required for the X chromosome.

Finally, an interval estimate of the location of a QTL may be obtained as the 1.5-LOD support interval: the region in which the LOD score is within 1.5 units of its chromosome-wide maximum. An approximate Bayes credible interval may also be used.

4.9 Further reading

Soller *et al.* (1976) were among the first to clearly describe the use of marker regression. Lander and Botstein (1989) is the seminal paper on interval mapping. The initial paper on the EM algorithm in general (and which coined the term) was Dempster *et al.* (1977).

Haley and Knott (1992) and Martínez and Curnow (1992) independently developed the method we now call Haley–Knott regression. (The discussion in Haley and Knott (1992) is more clear.) Whittaker *et al.* (1996) described a nice trick for Haley–Knott regression: regression of the phenotype on each pair of adjacent markers can give the same information as regression on the conditional QTL genotype probabilities at steps through the interval. This strategy can further reduce computational effort, but requires complete marker genotype data (which is seldom available), and one cannot allow for the presence of genotyping errors.

Feenstra *et al.* (2006) described the extended Haley–Knott method; a similar method was proposed by Xu (1998), though the iteratively reweighted least squares algorithm presented there is not quite right and can give negative LOD scores. The multiple imputation approach was proposed by Sen and Churchill (2001).

Lander and Botstein (1989) estimated significance thresholds for a genome scan by computer simulation as well as analytic means (the latter for the case of an infinitely dense map of markers). Churchill and Doerge (1994) proposed the use of permutation tests in this context. Manichaikul *et al.* (2007) described the use of a stratified permutation test in the presence of selective genotyping.

Broman *et al.* (2006) described the treatment of the X chromosome as presented in this chapter.

Lander and Botstein (1989) proposed the use of 1- and 2-LOD support intervals. Dupuis and Siegmund (1999) provided some support for the use of 1.5-LOD support intervals. Sen and Churchill (2001) described the Bayes intervals. Visscher *et al.* (1996b) described the use of the bootstrap in this context, though Manichaikul *et al.* (2006) showed that it behaves badly, and so the LOD support intervals and especially the Bayes credible intervals are preferred.

Beavis (1994) was the first to raise the issue of selection bias in estimated QTL effects (now often called the Beavis effect); see also Broman (2001).

Jiang and Zeng (1995) provide a good discussion of the joint analysis of multiple phenotypes.