

Non-normal phenotypes

The methods discussed in Chap. 4 all rely on the assumption that, given QTL genotype, the phenotype follows a normal distribution. This is not the same as to assume that the marginal phenotype distribution is normal—it will follow a mixture of normal distributions. But in the case that no QTL has very large effect, the marginal phenotype distribution would generally be close to normal: unimodal and reasonably symmetric.

While the normality assumption is often reasonable, departures from normality are not uncommon: the phenotype may be dichotomous, highly skewed, or exhibit spikes. (For example, if the phenotype is the mass of gallstones, some individuals may have no gallstones, and so a spike at 0 would be observed.) In practice, application of standard interval mapping will generally give reasonable results, even for a dichotomous trait, provided that statistical significance is established via a permutation test, and except for the problem of spurious LOD scores in regions of low genotype information (see Sec. 4.2.5).

Nevertheless, improved efficiency may be obtained by applying alternate methods. The simplest approach is to transform the phenotype. (For example, for the `iron` data in Sec. 4.4.3, we used a log transformation.) We generally stick to either taking logs, square roots, or no transformation. In this chapter, we describe several alternative interval mapping methods, including nonparametric interval mapping (based on the ranks of the phenotypes), interval mapping specific for binary traits, and a two-part model for the case of a phenotype distribution exhibiting a spike (such as at 0).

We conclude the chapter with a section describing, for the especially computer-savvy reader, how one can implement one's own QTL mapping method in R/qtl. This is illustrated by an implementation of a Cox proportional hazards model for right-censored phenotypes, using the Haley–Knott regression approach.

5.1 Nonparametric interval mapping

In the case of complete genotype data at a putative QTL, standard interval mapping is equivalent to using a t test (for a backcross) or analysis of variance (for an intercross). The nonparametric analogs of these methods are the Wilcoxon rank-sum test and the Kruskal–Wallis test. Here we describe the extension of these rank-based methods for interval mapping, in which the QTL genotypes are not known but must be inferred on the basis of multipoint marker genotype data. We will focus on the extension of the Kruskal–Wallis test, in which there is an arbitrary number of genotype groups; the Wilcoxon rank-sum test corresponds to the special case of two groups.

Let y_i denote the phenotype of individual i , and rank the phenotypes from $1, \dots, n$, with R_i denoting the rank for individual i . In the case of ties, one may randomize the ranks for any tied phenotypes, though we prefer to assign the average rank within each group of ties.

Consider some fixed position in the genome as the location of a putative QTL, and let $p_{ij} = \Pr(g_i = j | \mathbf{M}_i)$, the QTL genotype probabilities given the available multipoint marker data, $\mathbf{M}_i$. Whereas in the Kruskal–Wallis test statistic, one considers the sum of the ranks within each group, here the exact assignment of individuals to QTL genotype groups is not known; rather, individual i has prior probability p_{ij} of belonging to group j . Thus we consider the expected rank-sum, $S_j = \sum_i p_{ij} R_i$.

We then form the statistic

$$H = \sum_j \left(\frac{n - \sum_i p_{ij}}{n} \right) \left[\frac{(S_j - E_{0j})^2}{V_{0j}} \right]$$

where E_{0j} and V_{0j} are the mean and variance of S_j under the null hypothesis of no linkage, considering the p_{ij} as fixed. That is, E_{0j} and V_{0j} are the average and variance of the S_j if we take the R_i to be a random permutation of the integers $1, \dots, n$. We seek loci for which the expected rank sums, S_j , deviate from their average under the null hypothesis of no linkage.

After some algebra, we obtain the following formula.

$$H = \frac{12}{n(n+1)} \sum_j \frac{(n - \sum_i p_{ij}) (\sum_i p_{ij})^2}{n \sum_i p_{ij}^2 - (\sum_i p_{ij})^2} \left[\frac{\sum_i p_{ij} R_i}{\sum_i p_{ij}} - \frac{n+1}{2} \right]^2$$

In the case that the putative QTL is at a fully typed genetic marker, the p_{ij} will all be 0 or 1, and the above statistic reduces to the Kruskal–Wallis test statistic.

A standard correction for the case of ties is to use the statistic $H' = H/D$ where $D = 1 - \sum_k (t_k^3 - t_k)/(n^3 - n)$, with t_k being the number of values in the k th group of ties. If there are no ties, $D = 1$ and so $H' = H$. In the presence of ties, the correction results in a slight inflation of the test statistic. Note, however, that if a permutation test is used to establish statistical significance,

the correction factor is immaterial, as it will apply uniformly to the observed statistics as well as those from each permutation.

As the nonparametric statistic H' follows, approximately, a χ^2 distribution under the null hypothesis of no linkage, we convert the statistic to the LOD scale by taking $\text{LOD} = H'/(2 \ln 10)$. The resulting statistic is not a true LOD score, as a LOD score is a $\log_{10}$ likelihood ratio, and the nonparametric interval mapping method does not involve a likelihood. However, this transformation gives statistics whose values are more in line with those from standard interval mapping.

It is important to emphasize that this method for extending rank-based test statistics for the case of missing genotype information is more in the style of Haley–Knott regression than of standard interval mapping. In the case of appreciable missing genotype information, the method may lose efficiency. Thus, an alternate approach deserves mention: one might convert the ranked phenotypes to quantiles of the normal distribution and apply standard interval mapping. In other words, we could use as our phenotypes $z_i = \Phi^{-1}[(R_i - 1/2)/n]$, where Φ^{-1} is the inverse of the cumulative distribution function of the standard normal distribution. (That is, if Z follows a normal distribution with mean 0 and SD 1, $\Phi(z) = \Pr(Z \leq z)$, and Φ^{-1} is the inverse of this function.) This approach will not work well in the case of many ties in the phenotypes, particularly for the case of a spike in the phenotype distribution, but otherwise should give results similar to the nonparametric interval mapping method described above.

Example

To illustrate these nonparametric methods, we consider the `listeria` data, described in detail in Sec. 2.3 and 2.4. This is a mouse intercross; the phenotype concerns survival time following infection with *Listeria monocytogenes*, and exhibits a spike at 264 hours: approximately 30% of the mice survived past the 240-hour time point and were considered to have recovered from the infection; their phenotype was recorded as 264. (For a histogram, see the lower left panel in Fig. 2.7 on page 35.)

We first need to get access to the data (which are distributed with the R/qtl package), and run `calc.genoprob` to calculate the QTL genotype probabilities given the available marker genotype data.

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

Nonparametric interval mapping is performed with the `scanone` function, using the argument `model="np"`, as follows. (In `scanone`, the argument `model` refers to the phenotype model, by default taken to be `"normal"`, and the argument `method` refers to the analysis method. The `method` argument is ignored when `model="np"`.)

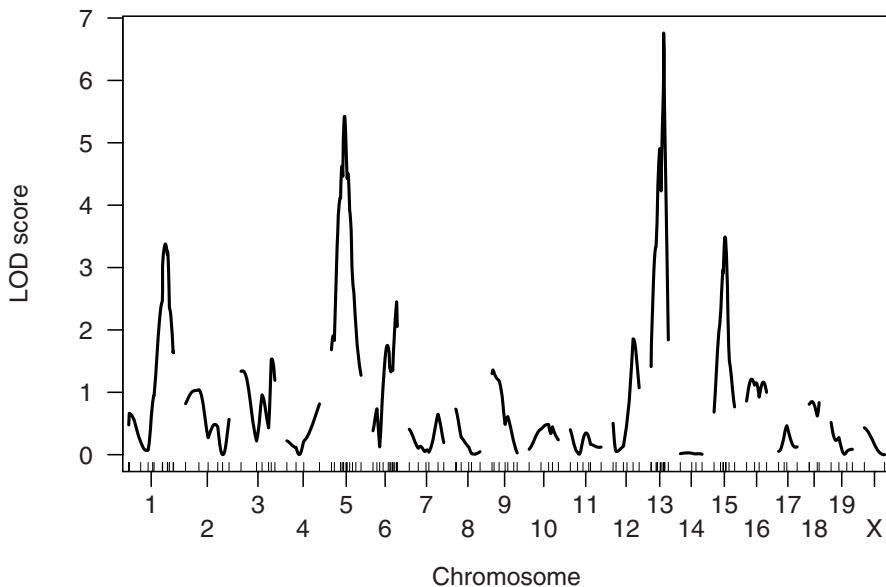


Figure 5.1. LOD scores by nonparametric interval mapping for the *listeria* data.

```
> out.np <- scanone(listeria, model="np")
```

By default, ties in the phenotypes are replaced by the average rank in each group of ties. With the argument `ties.random=TRUE`, ranks for tied phenotypes are randomized.

We can plot the results for all chromosomes as follows; the results appear in Fig. 5.1. We use `alternate.chrid=TRUE` so that the chromosome IDs may be more easily distinguished.

```
> plot(out.np, ylab="LOD score", alternate.chrid=TRUE)
```

As described in Chap. 4, a permutation test may be performed via the `n.perm` argument to `scanone`, as follows. We need to use `perm.Xsp=TRUE` to perform X-chromosome-specific permutations.

```
> operm.np <- scanone(listeria, model="np", n.perm=1000,
+                     perm.Xsp=TRUE)
```

The 5% LOD thresholds are the following.

```
> summary(operm.np, 0.05)
```

Autosome LOD thresholds (1000 permutations)

```
lod
5% 3.20
```

X chromosome LOD thresholds (25078 permutations)

```
lod
5% 2.33
```

Significant evidence for a QTL is seen on chromosomes 1, 5, 13, and 15.

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

	chr	pos	lod	pval
c1.loc76	1	76.0	3.38	0.0343
c5.loc27	5	27.0	5.42	0.0000
D13M147	13	26.2	6.76	0.0000
c15.loc23	15	23.0	3.49	0.0187

5.2 Binary traits

In human linkage analysis, much of the focus has been on binary traits, and methods for quantitative traits were developed later. With experimental crosses, on the other hand, researchers have focused almost exclusively on quantitative traits. Nevertheless, interval mapping for binary traits is no more difficult than for quantitative traits.

Let the phenotypes y_i take values either 0 or 1. (For example, assign unaffected individuals the value 0 and affected individuals 1.) Consider some fixed position in the genome as the location of a putative QTL, and let g_i denote the QTL genotype of individual i . Again, let $p_{ij} = \Pr(g_i = j | \mathbf{M}_i)$.

Let $\pi_j = \Pr(y_i = 1 | g_i = j)$, the penetrance of QTL genotype j . Given the marker data, $\mathbf{M}_i$, but not knowing the QTL genotypes g_i , the y_i follow mixtures of Bernoulli distributions (analogous to the mixtures of normal distributions that arise in standard interval mapping).

The likelihood for the parameters $\boldsymbol{\pi} = (\pi_j)$ is then

$$L(\boldsymbol{\pi}) = \prod_i \sum_j p_{ij} (\pi_j)^{y_i} (1 - \pi_j)^{(1-y_i)}$$

We obtain maximum likelihood estimates (MLEs), $\hat{\pi}_j$, using a form of the EM algorithm. At iteration $s + 1$, we have estimates of the parameters, $\hat{\boldsymbol{\pi}}^{(s)}$. In the E-step, we calculate weights for each individual and for each genotype:

$$w_{ij}^{(s+1)} = \Pr(g_i = j | y_i, \mathbf{M}_i, \hat{\boldsymbol{\pi}}^{(s)}) = \frac{p_{ij} (\hat{\pi}_j^{(s)})^{y_i} (1 - \hat{\pi}_j^{(s)})^{(1-y_i)}}{\sum_k p_{ik} (\hat{\pi}_k^{(s)})^{y_i} (1 - \hat{\pi}_k^{(s)})^{(1-y_i)}}.$$

In the M-step, we reestimate the probabilities π_j as weighted proportions using the weights, $w_{ij}^{(s+1)}$:

$$\hat{\pi}_j^{(s+1)} = \frac{\sum_i y_i w_{ij}^{(s+1)}}{\sum_i w_{ij}^{(s+1)}}.$$

We begin the algorithm by taking $w_{ij}^{(0)} = p_{ij}$, and iterate until the estimates converge, giving the MLE, $\hat{\pi}$.

We next calculate a LOD score for the test of $H_0 : \pi_j \equiv \pi$. First note that the MLE, under H_0 , of the common probability π is the overall proportion, $\hat{\pi}_0 = \sum_i y_i / n$. Letting $\hat{\pi}_0 = (\hat{\pi}_0, \hat{\pi}_0, \hat{\pi}_0)$, the LOD score is $\text{LOD} = \log_{10}\{L(\hat{\pi})/L(\hat{\pi}_0)\}$.

As with standard interval mapping, the likelihood under H_0 is calculated once, while the EM algorithm is performed at each position on a grid covering the genome, producing a LOD curve for each chromosome.

Example

Let us apply the binary trait method to the `listeria` data, taking as the binary trait whether the individuals' survived the infection or not. We first create the binary trait and append it to the phenotype data. Note that the function `pull.pheno` can be used to pull out a phenotype column.

```
> binphe <- as.numeric(pull.pheno(listeria, 1) > 250)
> listeria$pheno <- cbind(listeria$pheno, binary=binphe)
```

We need the results of `calc.genoprob`, but these were obtained in the previous section. The binary trait mapping method is performed with `scanone`, using the argument `model="binary"`. The phenotype must have values 0 and 1. We use the argument `pheno.col` to indicate that the new phenotype (named "binary") is to be used; we could also use `pheno.col=3` or even `pheno.col=binphe`.

```
> out.bin <- scanone(listeria, pheno.col="binary",
+                   model="binary")
```

We can plot the results for all chromosomes, together with the results obtained by the nonparametric method, as follows. The results appear in Fig. 5.2.

```
> plot(out.np, out.bin, col=c("blue", "red"), ylab="LOD score",
+      alternate.chrid=TRUE)
```

A permutation test is again performed using the `n.perm` argument to `scanone`. Again we should use `perm.Xsp=TRUE`.

```
> operm.bin <- scanone(listeria, pheno.col="binary",
+                     model="binary", n.perm=1000,
+                     perm.Xsp=TRUE)
```

The LOD thresholds for the 5% significance level are the following.

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

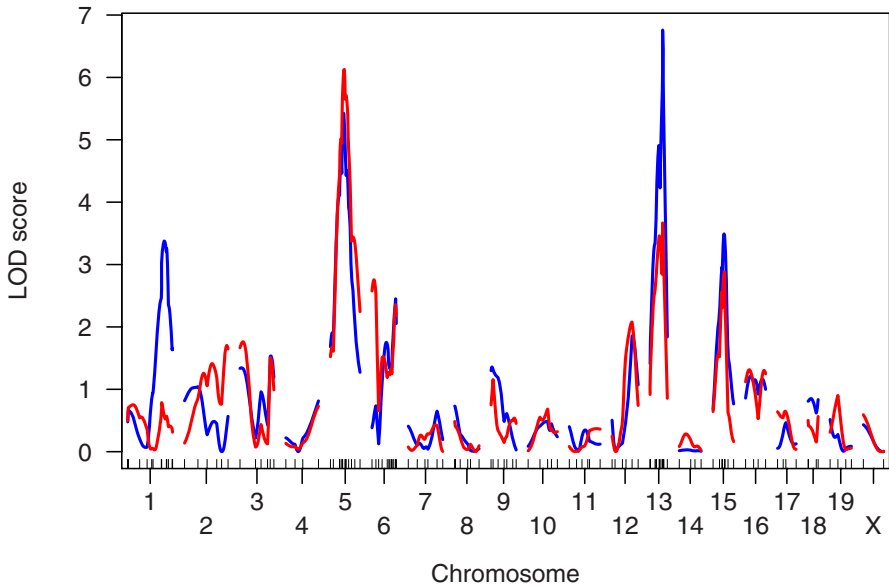


Figure 5.2. LOD scores by nonparametric interval mapping (in blue) and binary trait mapping (in red) for the *listeria* data. The binary trait is defined by survival > 250 hr or not.

Autosome LOD thresholds (1000 permutations)

lod

5% 3.63

X chromosome LOD thresholds (25078 permutations)

lod

5% 2.53

Significant evidence for a QTL is seen on chromosomes 5 and 13.

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

	chr	pos	lod	pval
c5.loc29	5	29.0	6.13	0.00104
D13M147	13	26.2	3.67	0.04468

5.3 Two-part model

One often observes a spike in the phenotype distribution, such as that observed for the survival phenotype in the *listeria* data. Another common example

is the case of mass of tumor, with some individuals exhibiting no tumor. In this section, we describe an analysis method particular for this situation.

Assume, without loss of generality, that the spike in the phenotype distribution is at 0. Let y_i denote the quantitative phenotype for individual i . Let $z_i = 0$ if $y_i = 0$, and $z_i = 1$ if $y_i > 0$.

A simple approach for QTL mapping in this situation is to first analyze the quantitative phenotype, y_i , using only the individuals for which $y_i > 0$, by standard interval mapping, and then separately analyze the binary trait z_i . These can be combined in what we call the two-part model.

Assume that the $(\mathbf{M}_i, y_i, z_i)$ are mutually independent, that $\Pr(z_i = 1 | g_i = j) = \pi_j$, and that $y_i | (g_i = j, z_i = 1) \sim \text{normal}(\mu_j, \sigma^2)$. In other words, the probability that an individual with QTL genotype j has the null phenotype is $1 - \pi_j$; if this individual's phenotype is non-null, it follows a normal distribution with mean μ_j , depending on the QTL genotype, and with SD σ , independent of genotype.

In an intercross, this model contains seven parameters, $\boldsymbol{\theta} = (\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3, \sigma)$. The likelihood function is the following:

$$L(\boldsymbol{\theta}) = \prod_i \sum_j p_{ij} (1 - \pi_j)^{1-z_i} \{\pi_j \phi(y_i; \mu_j, \sigma)\}^{z_i}$$

where $\phi(y; \mu, \sigma)$ is the density function for a normal distribution with mean μ and SD σ .

We may again obtain MLEs with a form of the EM algorithm. Assume at iteration $s + 1$ we have estimates $\hat{\boldsymbol{\theta}}^{(s)}$. In the E-step, we calculate weights for each individual and each genotype:

$$w_{ij}^{(s+1)} = \Pr(g_i = j | y_i, z_i, \mathbf{M}_i, \hat{\boldsymbol{\theta}}^{(s)}) = \begin{cases} \frac{p_{ij} (1 - \hat{\pi}_j^{(s)})}{\sum_k p_{ik} (1 - \hat{\pi}_k^{(s)})} & \text{if } z_i = 0 \\ \frac{p_{ij} \hat{\pi}_j^{(s)} \phi(y_i; \hat{\mu}_j^{(s)}, \hat{\sigma}^{(s)})}{\sum_k p_{ik} \hat{\pi}_k^{(s)} \phi(y_i; \hat{\mu}_k^{(s)}, \hat{\sigma}^{(s)})} & \text{if } z_i = 1 \end{cases}$$

In the M-step, we obtain revised estimates of the parameters according to the following equations:

$$\begin{aligned} \hat{\pi}_j^{(s+1)} &= \frac{\sum_i w_{ij}^{(s+1)} z_i}{\sum_i w_{ij}^{(s+1)}} \\ \hat{\mu}_j^{(s+1)} &= \frac{\sum_i y_i w_{ij}^{(s+1)} z_i}{\sum_i w_{ij}^{(s+1)} z_i} \\ \hat{\sigma}^{(s+1)} &= \sqrt{\frac{\sum_i \sum_j (y_i - \hat{\mu}_j^{(s+1)})^2 w_{ij}^{(s+1)} z_i}{\sum_i z_i}} \end{aligned}$$

We again start the algorithm by taking $w_{ij}^{(0)} = p_{ij}$, and iterate until the estimates converge, producing the MLEs, $\hat{\theta}$.

We may calculate a LOD score for the test of $H_0 : \pi_j \equiv \pi, \mu_j \equiv \mu$. We first note that, under H_0 , the MLEs of the three parameters, π , μ , and σ , are

$$\begin{aligned}\hat{\pi}_0 &= \frac{\sum_i z_i}{n} \\ \hat{\mu}_0 &= \frac{\sum_i z_i y_i}{\sum_i z_i} \\ \hat{\sigma}_0 &= \sqrt{\frac{\sum_i (y_i - \hat{\mu}_0)^2 z_i}{\sum_i z_i}}.\end{aligned}$$

In other words, $\hat{\pi}_0$ is the proportion of individuals with a positive phenotype, and $\hat{\mu}_0$ and $\hat{\sigma}_0$ are the sample mean and SD, among individuals with positive phenotypes. Letting $\hat{\theta}_0 = (\hat{\pi}_0, \hat{\pi}_0, \hat{\pi}_0, \hat{\mu}_0, \hat{\mu}_0, \hat{\mu}_0, \hat{\sigma}_0)$, the LOD score is $\text{LOD} = \log_{10}\{L(\hat{\theta})/L(\hat{\theta}_0)\}$.

We calculate two additional sets of LOD scores. First, we consider the hypothesis $H'_0 : \pi_j \equiv \pi$, but allowing the μ_j to vary; the corresponding LOD scores assess evidence for QTL that specifically influence the chance that an individual has the null phenotype. Second, we consider the hypothesis $H''_0 : \mu_j \equiv \mu$, but allowing the π_j to vary; the corresponding LOD scores assess evidence for QTL that influence the average phenotype, among individuals with a non-null phenotype. We could say that H'_0 concerns the penetrance of the disease and H''_0 concerns the severity of the disease.

Note that in the case of complete QTL genotype information (i.e., when the putative QTL is at a marker that has been fully typed), the p_{ij} are all either 1 or 0, and the two parts of the model fully separate. In this case, the MLEs under the two-part model are exactly those obtained by the two separate analyses (the analysis of the binary trait and the conditional analysis of the quantitative trait, for those individuals with nonzero phenotype). Further, the LOD score for the two-part model is simply the sum of the LOD scores from the two separate analyses.

Example

As an illustration, we again consider the `listeria` data. Analysis with the two-part model is performed with `scanone` using `model="2part"`. The spike in the phenotype is assumed to be either the largest or the smallest observed phenotype. By default, the smallest phenotype is assumed; for the `listeria` data we must use the argument `upper=TRUE` to indicate that it is the largest observed phenotype (264 hr) that is to be treated as the spike. We will consider log survival time as the phenotype, and so we first append log survival to the phenotype data.

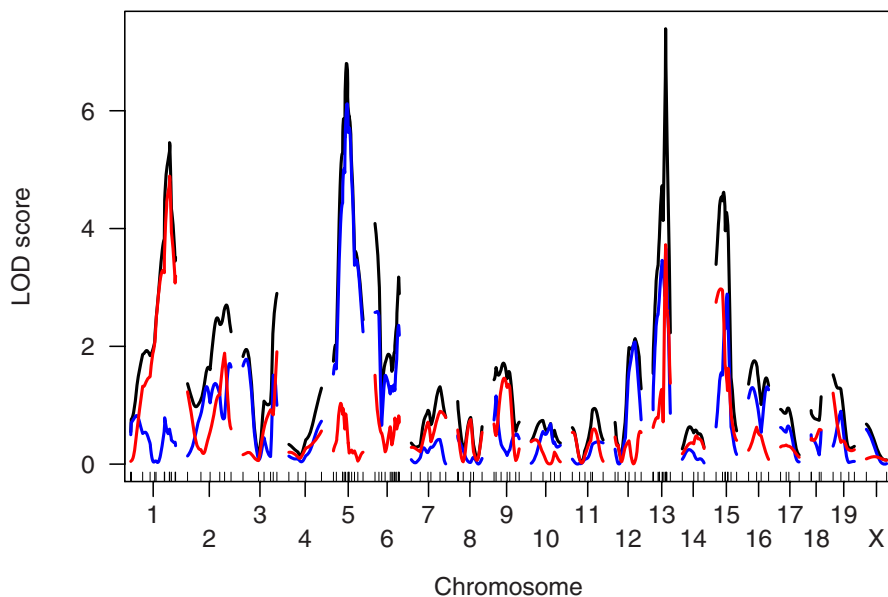


Figure 5.3. LOD scores from the two-part model, with $\text{LOD}(\pi, \mu)$ in black, $\text{LOD}(\pi)$ in blue and $\text{LOD}(\mu)$ in red, for the *listeria* data.

```
> y <- log(pull.pheno(listeria, 1))
> listeria$pheno <- cbind(listeria$pheno, logsurv=y)
```

We now use `scanone` to calculate the LOD curves.

```
> out.2p <- scanone(listeria, model="2part", upper=TRUE,
+                   pheno.col="logsurv")
```

The results (see Fig. 5.3) contain three LOD score columns: $\text{LOD}(\pi, \mu)$, for the overall test of $\pi_j \equiv \pi$ and $\mu_j \equiv \mu$; $\text{LOD}(\pi)$, for the test of $\pi_j \equiv \pi$; and $\text{LOD}(\mu)$, for the test of $\mu_j \equiv \mu$. We plot all three together as follows. The argument `lodcolumn` is used to plot all three LOD score columns.

```
> plot(out.2p, lodcolumn=1:3, ylab="LOD score",
+       alternate.chrid=TRUE)
```

The results indicate that the locus on chromosome 1 largely affects time-to-death, given that an individual has died ($\text{LOD}(\mu)$, in red, is large while $\text{LOD}(\pi)$, in blue, is small). The locus on chromosome 5 largely affects the chance of survival ($\text{LOD}(\pi)$, in blue, is large while $\text{LOD}(\mu)$, in red, is small). The loci on chromosomes 13 and 15 affect both aspects of the phenotype (both $\text{LOD}(\mu)$ and $\text{LOD}(\pi)$ are large).

A permutation test is performed as follows.

```
> operm.2p <- scanone(listeria, model="2part", upper=TRUE,
+                      pheno.col="logsurv", n.perm=1000,
+                      perm.Xsp=TRUE)
```

The results, for each of the autosomes and X chromosome, contain 3 columns: the genome-wide maxima of $\text{LOD}(\pi, \mu)$, $\text{LOD}(\pi)$, and $\text{LOD}(\mu)$, for each permutation replicate. We obtain LOD thresholds as follows. (Note that π is denoted p in the output.)

```
> summary(operm.2p, alpha=0.05)
```

Autosome LOD thresholds (1000 permutations)

```
lod.p.mu lod.p lod.mu
5%      4.8  3.63  3.88
```

X chromosome LOD thresholds (25078 permutations)

```
lod.p.mu lod.p lod.mu
5%      3.18  2.54  2.58
```

If we consider just the overall LOD score, $\text{LOD}(\pi, \mu)$, significant evidence for a QTL is seen on chromosomes 1, 5, and 13.

```
> summary(out.2p, perms=operm.2p, alpha=0.05, pvalues=TRUE)
```

	chr	pos	lod.p.mu	pval	lod.p	pval	lod.mu
c1.loc81	1	81.0	5.46	0.02183	0.594	1.00000	4.890
c5.loc27	5	27.0	6.80	0.00416	6.030	0.00104	0.779
D13M147	13	26.2	7.39	0.00312	3.667	0.04571	3.726
				pval			
c1.loc81				0.00832			
c5.loc27				1.00000			
D13M147				0.06750			

As discussed in Sec. 4.7, we may use the `format` argument of `summary.scanone` to get the maximum LOD scores for each of the three LOD score columns in `out.2p`.

```
> summary(out.2p, perms=operm.2p, alpha=0.05, pvalues=TRUE,
+          format="allpeaks")
```

	chr	pos	lod.p.mu	pval	pos	lod.p	pval	pos	lod.mu
1	1	81.0	5.46	0.02183	12.0	0.825	1.00000	81.0	4.89
5	5	27.0	6.80	0.00416	29.0	6.117	0.00104	15.0	1.03
13	13	26.2	7.39	0.00312	26.2	3.667	0.04571	26.2	3.73
				pval					
1				0.00832					
5				1.00000					
13				0.06750					

Note that the locus on chromosome 15 did not achieve the 5% significance level here, but was seen to have a significant effect by nonparametric interval mapping (see Sec. 5.1). The linkage test in nonparametric interval mapping concerns two degrees of freedom, while with the two-part model, the test concerns four degrees of freedom. Thus, the two-part model has a higher significance threshold and so lower power for QTL detection. On the other hand, the separation of penetrance and severity can give a more detailed understanding of the QTL effects.

5.4 Other extensions

Essentially any phenotype model that is of interest in linear regression would also find application for QTL mapping. Only a limited number have been implemented in R/qtl. Thus, we conclude this chapter with a description, for the more computationally-savvy reader, of how such alternative mapping methods may be implemented with R/qtl.

We consider, as an illustration, the case of right-censored phenotypes. A phenotype is right-censored if it is only known to be greater than some value. For example, in the *listeria* data, a large proportion of individuals had not died of the *Listeria monocytogenes* infection at 264 hr. In the two-part model described in the previous section, these were viewed as having recovered from the infection, but we could also view the outcome as a survival time, and that the survival time for these individuals was right-censored.

A common approach for the analysis of such data is the use of a Cox proportional hazards model. Consider a random survival time, Y , and let $f(y)$ denote its density and $S(y) = \Pr(Y > y)$ denote its survival function. The *hazard function* is $h(y) = f(y)/S(y)$, which is essentially the chance that an individual will die immediately at time y given that it has survived to that point.

In the Cox proportional hazards model, we assume that the hazard function for individuals with QTL genotype g is $h_g(y) = h_0(y)e^{\beta_g}$, where h_0 is some completely unspecified baseline hazard function, and the β_g are the effects of the QTL genotypes. While the QTL genotypes will generally not be known, we may use an approach analogous to Haley–Knott regression. Let $p_{ij} = \Pr(g_i = j | \mathbf{M}_i)$ denote the QTL genotype probabilities given the available marker data, and assume that the hazard function for individual i is $h_0(y) \exp[\sum_j \beta_j p_{ij}]$.

To fit the Cox proportional hazards model, we use the **survival** package, which is distributed with R.

A function to perform the analysis appears in Fig. 5.4. As described in Sec. 4.4, the X chromosome requires special treatment, but we will just omit it from consideration here.

In line 4, we use the **require** function to ensure that the **survival** package is loaded. In lines 6–10 we omit individuals with missing phenotypes and make

```

1  scanone.cph <-
   function(cross, pheno.col=1)
   {
       require(survival)

5      pheno <- pull.pheno(cross, pheno.col)
       cross <- subset(cross, ind=!is.na(pheno))
       pheno <- pheno[!is.na(pheno)]
       if(class(pheno) != "Surv")
10      stop("Need the phenotype to be of class \"Surv\".")

       chrtype <- sapply(cross$geno, class)
       if(any(chrtype=="X")) {
           warning("Dropping X chromosome.")
15      cross <- subset(cross, chr=(chrtype != "X"))
       }
       chr <- names(cross$geno)

       result <- NULL
20      for(i in 1:nchr(cross)) {
           if(!("prob" %in% names(cross$geno[[i]]))) {
               warning("First running calc.genoprob.")
               cross <- calc.genoprob(cross)
           }
25      p <- cross$geno[[i]]$prob

           # pull out map; drop last column of probabilities
           map <- attr(p, "map")
           p <- p[,,-dim(p)[3],drop=FALSE]

30      lod <- apply(p, 2, function(a,b)
                   diff(coxph(b ~ a)$loglik)/log(10), pheno)

           z <- data.frame(chr=chr[i], pos=map, lod=lod)

35      # special names for rows
       w <- names(map)
       o <- grep("^loc-*[0-9]+", w)
       if(length(o) > 0) # locations cited as "c*.loc*"
40      w[o] <- paste("c",names(cross$geno)[i],".",w[o],sep="")
       rownames(z) <- w

       result <- rbind(result, z)
   }
45  class(result) <- c("scanone", "data.frame")
   result
}
```

Figure 5.4. The scanone.cph function.

sure that the phenotype has been appropriately converted to be a survival time (see below). In lines 12–17, we omit the X chromosome and store the names of the chromosomes.

In line 19, we create a dummy object that will contain all of the results. In line 20, we begin a loop over the chromosomes. In lines 21–25 we check that the results of `calc.genoprob` are available, and pull out the probabilities for the current chromosome.

In line 28, we pull out the genetic map for the grid on which the QTL genotype probabilities were calculated. In line 29, we drop the last column from the QTL genotype probabilities, since the analysis will include an intercept term.

In lines 31–32, we perform the actual analysis. We use the `apply` function to send the QTL genotypes, one column at a time, to the `coxph` function. The `coxph` function is part of the `survival` package, and performs the Cox proportional hazards regression. Part of the output of `coxph` is the log (base e) likelihood, for the null model (with just the intercept) and for the alternative model (including the covariates: here, the QTL genotype probabilities). We take the difference of these log likelihoods and then divide by $\ln(10)$ to convert the result to the LOD scale.

In line 34, we paste together the chromosome IDs, map positions, and LOD scores into a data frame. In lines 36–41 we create special row names, used to ensure clarity regarding which positions are markers and which are between markers. In line 43, we append the results for this chromosome to the end of our growing set of results.

Finally, in line 45, we change the “class” of the result to include “`scanone`”, so that we may use `plot` and `summary` and have the data sent to `plot.scanone` and `summary.scanone`. The final line contains the return value for the function.

Example

Let us now apply the method to the `listeria` data. We first need to load the `survival` package and the code for the `scanone.cph` function; the latter may be done with the `source` function.

```
> library(survival)
> source("scanone_cph.R")
```

We need to convert the phenotype into a censored survival time, using the function `Surv` in the `survival` package. This is done to indicate which phenotypes are to be viewed as censoring times rather than actual survival times. `Surv` takes two arguments: the survival/censoring times and an indicator of whether the values were observed or were censored. We append this revised phenotype to the end of the phenotype data. (This will now be the fifth phenotype in the data set.)

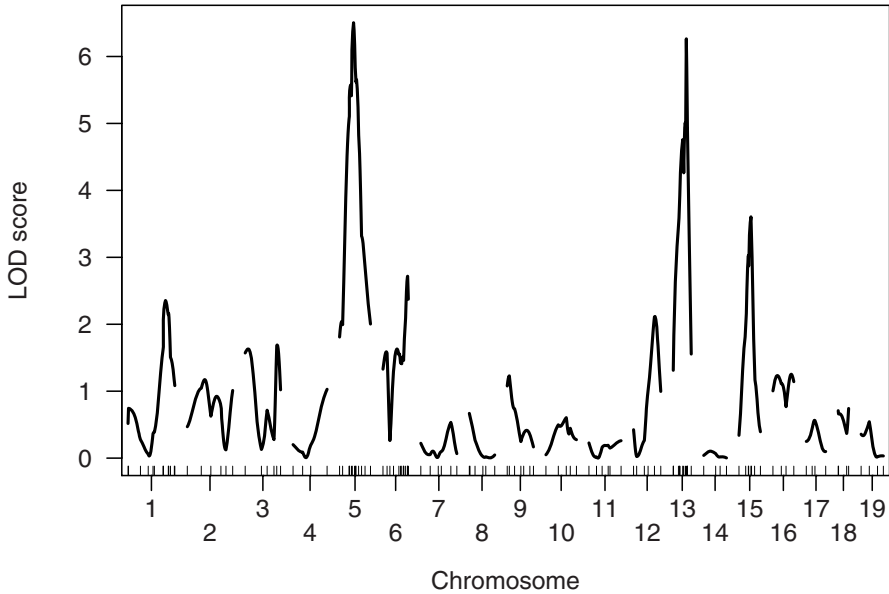


Figure 5.5. LOD scores from the Cox proportional hazards model, using an approach analogous to Haley–Knott regression, for the `listeria` data.

```
> y <- pull.pheno(listeria, 1)
> y <- Surv(y, y<250)
> listeria$pheno <- cbind(listeria$pheno, surv=y)
```

We may now send the data to `scanone.cph` to perform the analysis. We may refer to the phenotype by its name, "surv".

```
> out.cph <- scanone.cph(listeria, pheno.col="surv")
```

We may plot the resulting LOD scores as follows. (See Fig. 5.5.)

```
> plot(out.cph, ylab="LOD score", alternate.chrid=TRUE)
```

We have not written code to do a permutation test, and so we will perform the permutation test by brute force, using a `for` loop.

```
> n.perm <- 1000
> operm.cph <- cbind(lod=1:n.perm)
> chr <- names(listeria$geno)
> temp <- subset(listeria, chr!=(chr != "X"))
> n.ind <- nind(listeria)
> for(i in 1:n.perm) {
+   temp$pheno <- temp$pheno[sample(n.ind),]
+   out <- scanone.cph(temp, pheno.col="surv")
+   operm.cph[i] <- max(out[,3], na.rm=TRUE)
```

```
+ }
> class(operm.cph) <- "scanoneperm"
```

The LOD threshold for the 5% significance level is the following.

```
> summary(operm.cph, 0.05)

LOD thresholds (1000 permutations)
  lod
5% 3.51
```

Significant evidence for a QTL is seen on chromosomes 5, 13 and 15.

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

      chr pos lod pval
c5.1loc28    5 28.0 6.50 0.000
D13M147     13 26.2 6.26 0.000
c15.1loc24   15 24.0 3.60 0.039
```

5.5 Summary

With the interval mapping methods of Chap. 4, one assumes that the residual variation in the phenotype follows a normal distribution. While the normality assumption is often appropriate, many phenotypes show clear departures from normality. Application of interval mapping often performs reasonably anyway, particularly if the phenotype is transformed to give approximate normality. However, there are alternatives. Nonparametric interval mapping considers the ranks of the phenotype values. An interval mapping method specific for binary traits is simple to develop. In general, one may generate an interval mapping method tailored to any sort of phenotype model.

5.6 Further reading

The first three methods discussed in this chapter were considered in Broman (2003), in which the two-part model was proposed. Kruglyak and Lander (1995) proposed the nonparametric interval mapping approach for backcrosses; they described a somewhat different method for dealing with intercrosses, but we prefer the extension of the Kruskal–Wallis test statistic. Xu and Atchley (1996) described the approach for binary traits, though in a somewhat more general context that included marker covariates. Visscher *et al.* (1996a) and McIntyre *et al.* (2001) both described approximate methods for interval mapping with binary traits, but these are not recommended.

Several authors have described QTL mapping methods for survival times. Symons *et al.* (2002) used a Monte Carlo method to fit a semiparametric

Cox proportional hazards model (Cox, 1972), making more precise use of the genotype data than the method described in Sec. 5.4. Diao and Lin (2005) considered the same model, but used a different, and less computationally demanding method for estimation. Diao *et al.* (2004) described maximum likelihood for a fully parametric Weibull model. Moreno *et al.* (2005) studied the performance of the Weibull and Cox proportional hazards model relative to standard interval mapping.

Several other methods deserve mention, but have not yet been implemented in R/qtl. Hackett and Weller (1995) described a method for the analysis of ordinal traits. Jansen (1992, 1993b) described the use of generalized linear models for QTL mapping, which could be applied for a variety of types of phenotypes, including count data.