

Karl W. Broman, Saunak Sen

A guide to QTL mapping  
with R/qtl

Springer

---

## Preface

QTL are quantitative trait loci: genetic loci that contribute to variation in a quantitative trait. QTL mapping is the effort to identify QTL through an experimental cross.

In this book, we give an overview of the practical aspects of the analysis of QTL mapping experiments based on inbred line crosses, with explicit instructions on the use of the R/qrtl software (an add-on package for the general statistical software, R). We give some of the details of the statistical methods, but we mostly focus on how to get and make sense of results. Real data examples are included throughout.

The intended audience includes scientists who are performing QTL mapping experiments and participating directly in the analysis. We expect the reader to have a general understanding of statistical methods, including maximum likelihood estimation and linear regression. Some readers will be statisticians analyzing data from QTL experiments with a basic understanding of genetics. We provide limited introduction to either statistics or genetics. Readers with a limited understanding of statistics may wish to first study Rice (2006). Readers with a limited understanding of genetics may wish to first study Brown (2006). Alternatively, one might consider *The Cartoon Guide to Statistics* (Gonick and Smith, 1993) and *The Cartoon Guide to Genetics* (Gonick and Wheelis, 1991), which are more gentle and entertaining (but less complete) introductions to the subjects.

In line with our aim to describe the practical aspects of QTL mapping, the book contains extensive discussion of the R/qrtl software. We have attempted to separate the discussion of R/qrtl into subsections, so that readers who wish to focus on the basic ideas and skip over the software considerations may do so. In some places (e.g., Chap. 3, on data diagnostics), this was not feasible.

While much can be accomplished with R/qrtl (and much of this book may be read) with a limited understanding of R, efficient use of the software (and an understanding of more complex R/qrtl code) requires a more detailed understanding of R. We provide very little discussion of R itself, and refer the

reader to Dalgaard (2002), for a gentle introduction to R, and Venables and Ripley (2002), for a more comprehensive discussion of R.

The content of the book is ordered according to the way in which QTL analyses might proceed. (There is one exception: we postpone the discussion of experimental design to Chap. 6, as it requires a reasonably complete understanding of QTL mapping.) We begin with an introduction (Chap.1), including an overview of the structure of data from a QTL mapping experiment and the basic statistical problems. In Chap. 2, we explain how to import QTL mapping data into R/qlt, we describe some of the example data sets that will be considered further in later chapters, and we demonstrate how one may simulate QTL mapping data in R/qlt. At the end of the chapter, we describe the internal structure of QTL mapping data within R/qlt; this section should probably be skipped at first reading. In Chap. 3, we describe the various diagnostic procedures for assessing the quality and integrity of QTL mapping data.

Chapter 4 is the heart of the book. There, we discuss the basic approach to QTL mapping (interval mapping), the assessment of statistical significance in a genome scan, and the calculation of confidence intervals for QTL location. We focus on the case that residual variation in the phenotype follows a normal distribution. In Chap. 5, we consider several extensions of standard interval mapping for non-normal phenotypes.

In Chap. 6, we describe various experimental design issues, including the choice of cross, marker density, and sample size, and selective genotyping strategies. We consider both the power to detect a QTL and the precision of localization of QTL. We focus on the use of the R/qltDesign software (another add-on package for R), but also describe how one may estimate power and precision through computer simulation with R/qlt.

In Chap. 7, we describe the use of covariates in QTL mapping. We initially consider the inclusion of additive covariates (in which the effect of the QTL is constant, independent of the value of the covariate), but we also discuss the investigation of QTL  $\times$  covariate interactions. We conclude the chapter with a discussion of composite interval mapping (CIM), in which genetic markers are included as covariates.

The first seven chapters focus almost exclusively on single-QTL models. In Chap. 8, we take the first step towards multiple-QTL models by considering two-dimensional, two-QTL genome scans. Such two-dimensional scans offer the opportunity to assess evidence for linked or interacting QTL. In Chap. 9, we provide a more comprehensive discussion of the identification and exploration of multiple-QTL models. The problem is viewed as one of model selection in multiple linear regression, though with a number of special features.

We conclude the book with two case studies (Chap. 10 and 11), in order to illustrate the entirety of the process of mapping QTL. We bring together all of the tools discussed in the previous chapters to demonstrate their combined use in order to solve two moderately difficult problems.

The book has been written with a variety of possible readers in mind, including experienced QTL mappers interested in adopting the R/qtl software, postdoctoral researchers new to QTL mapping, and statistics graduate students interested in exploring applications of statistics. We do not expect that the book will be often read front-to-back in a linear fashion, and different readers will likely wish to approach the book differently.

The experienced QTL mapper might start with Chap. 2, on importing QTL mapping data sets, but would then likely skip about, making liberal use of the Contents and Index to identify sections of particular interest. The reader new to QTL mapping should start with the Introduction (Chap. 1), but might skip Chap. 2 and 3 at first reading and jump right into Chap. 4, in which the essentials of QTL mapping are described.

We have created a web site with on-line complements for the book (see <http://www.rqtl.org/book>). Included on that site are files with all of the R code used in the book, including the detailed code used to create the figures. We have also created an R package, R/qtlbook, containing all of our example data sets (except those already included in R/qtl).

We thank Victor Boyartchuk, Bill Dietrich, Mehmet Guler, Krista Nichols, Virginie Orgogozo, Sarah Owens, Bev Paigen, Karlyne Reilly, Noel Rose, Andy Smith, Michelle Southard-Smith, and Gary Thorgaard for providing data and for allowing its distribution. The public distribution of data is invaluable for statistical genetic methods development, and for learning. We further thank Aimee Teo Broman, Ken Manly, Krista Nichols, Virginie Orgogozo, Abraham Palmer, and several anonymous reviewers for suggestions to improve the book, and Sungjin Kim for identifying a number of typographical errors. Our ideas on QTL mapping were greatly influenced by Gary Churchill, Mark Neff, and Terry Speed; we thank them for many years of stimulating discussions. Our efforts were supported, in part, by NIH grants R01-GM074244 and R01-GM078338.

The book was created using R version 2.8.1, R/qtl version 1.11-12, R/qtl-Design version 0.92, and R/qtlbook version 0.16-3. Later versions of these software may have some minor differences; important changes will be described in the on-line complements (<http://www.rqtl.org/book>). The book was constructed with L<sup>A</sup>T<sub>E</sub>X and Sweave; we don't know how we could have done it otherwise. We thank the developers of R, L<sup>A</sup>T<sub>E</sub>X, and Sweave for making this work possible.

Madison, Wisconsin; San Francisco, California  
August, 2009

*Karl W. Broman*  
*Šaunak Sen*

---

## Contents

<b>1</b>	<b>Introduction</b>	1
1.1	Why perform a QTL experiment?	2
1.2	Crosses and data	3
1.2.1	Mouse hypertension data as an example	8
1.3	Central statistical problems	9
1.3.1	Models for recombination	12
1.3.2	Models connecting genotype and phenotype	14
1.4	About R and R/qt1	17
1.5	Other software	18
1.6	Work flow	19
1.7	Further reading	20
<b>2</b>	<b>Importing and simulating data</b>	21
2.1	Importing data	22
2.1.1	Comma-delimited files	22
2.1.2	MapMaker/QTL	30
2.1.3	QTL Cartographer	31
2.1.4	Map Manager QTX	32
2.2	Exporting data	32
2.3	Example data	33
2.4	Data summaries	34
2.5	Simulating data	36
2.5.1	Additive models	37
2.5.2	More complex models	40
2.6	Internal data structure	42
2.6.1	Experimental cross	42
2.6.2	Genetic map	45
2.7	Further reading	46

<b>3</b>	<b>Data checking</b> .....	47
3.1	Phenotypes .....	47
3.2	Segregation distortion .....	50
3.3	Compare individuals' genotypes .....	52
3.4	Check marker order .....	53
3.4.1	Pairwise recombination fractions .....	53
3.4.2	Rippling marker order .....	60
3.4.3	Estimate genetic map .....	64
3.5	Identifying genotyping errors .....	66
3.6	Counting crossovers .....	68
3.7	Missing genotype information .....	70
3.8	Summary .....	72
3.9	Further reading .....	73
<b>4</b>	<b>Single-QTL analysis</b> .....	75
4.1	Marker regression .....	75
4.2	Interval mapping .....	80
4.2.1	Standard interval mapping .....	80
4.2.2	Haley–Knott regression .....	86
4.2.3	Extended Haley–Knott regression .....	88
4.2.4	Multiple imputation .....	91
4.2.5	Comparison of methods .....	94
4.3	Significance thresholds .....	104
4.4	The X chromosome .....	108
4.4.1	Analysis .....	109
4.4.2	Significance thresholds .....	113
4.4.3	Example .....	114
4.5	Interval estimates of QTL location .....	118
4.6	QTL effects .....	122
4.7	Multiple phenotypes .....	127
4.8	Summary .....	131
4.9	Further reading .....	132
<b>5</b>	<b>Non-normal phenotypes</b> .....	135
5.1	Nonparametric interval mapping .....	136
5.2	Binary traits .....	139
5.3	Two-part model .....	141
5.4	Other extensions .....	146
5.5	Summary .....	150
5.6	Further reading .....	150

<b>6</b>	<b>Experimental design and power</b> .....	153
6.1	Phenotypes and covariates .....	153
6.2	Strains and strain surveys .....	154
6.3	Theory .....	155
6.3.1	Variance attributable to a locus .....	155
6.3.2	Residual error variance .....	157
6.3.3	Information content .....	158
6.4	Examples with R/qtlDesign .....	159
6.4.1	Functions .....	159
6.4.2	Choosing a cross .....	160
6.4.3	Genotyping strategies .....	164
6.4.4	Phenotyping strategies .....	166
6.4.5	Fine mapping .....	167
6.5	Other experimental populations .....	168
6.6	Estimating power and precision by simulation .....	170
6.7	Summary .....	176
6.8	Further reading .....	177
<b>7</b>	<b>Working with covariates</b> .....	179
7.1	Additive covariates .....	179
7.2	QTL $\times$ covariate interactions .....	190
7.3	Covariates with non-normal phenotypes .....	198
7.4	Composite interval mapping .....	205
7.5	Summary .....	210
7.6	Further reading .....	210
<b>8</b>	<b>Two-dimensional, two-QTL scans</b> .....	213
8.1	The normal model .....	214
8.2	Binary traits .....	228
8.3	The X chromosome .....	232
8.4	Covariates .....	236
8.5	Summary .....	239
8.6	Further reading .....	239
<b>9</b>	<b>Fit and exploration of multiple-QTL models</b> .....	241
9.1	Model selection .....	242
9.1.1	Class of models .....	244
9.1.2	Model fit .....	246
9.1.3	Model search .....	248
9.1.4	Model comparison .....	250
9.1.5	Further discussion .....	254

9.2	Bayesian QTL mapping	255
9.3	Multiple QTL mapping in R/qlt	258
9.3.1	makeqtl and fitqtl	259
9.3.2	refineqtl	263
9.3.3	addint	266
9.3.4	addqtl	267
9.3.5	addpair	269
9.3.6	Manipulating qtl objects	272
9.3.7	stepwiseqtl	274
9.4	Summary	281
9.5	Further reading	281
<b>10</b>	<b>Case study I</b>	<b>283</b>
10.1	Diagnostics	284
10.2	Initial cross	291
10.3	Combined data	300
10.4	Discussion	311
<b>11</b>	<b>Case study II</b>	<b>313</b>
11.1	Diagnostics	314
11.2	Initial QTL analyses	323
11.3	QTL $\times$ covariate interactions	339
11.4	Discussion	353
<b>A</b>	<b>Installing R and R/qlt</b>	<b>355</b>
A.1	Installing R	355
A.1.1	Windows	355
A.1.2	Mac OS X	356
A.1.3	Unix/Linux	356
A.2	Installing R/qlt	357
A.3	Optimizing the R environment	358
A.4	Working directories	358
A.5	Documentation	359
A.6	Email lists	360
<b>B</b>	<b>List of functions in R/qlt</b>	<b>361</b>
<b>C</b>	<b>QTL mapping data sets</b>	<b>365</b>
<b>D</b>	<b>Hidden Markov models for QTL mapping</b>	<b>371</b>
D.1	Specification of the model	372
D.1.1	The backcross	373
D.1.2	The intercross	374
D.2	QTL genotype probabilities	374
D.3	Simulation of QTL genotypes	376
D.4	Joint QTL genotype probabilities	377



D.5 The Viterbi algorithm .....	378
D.6 Estimation of intermarker distances .....	379
D.7 Detection of genotyping errors .....	380
D.8 A practical issue .....	381
D.9 Further reading .....	381
<b>References</b> .....	<b>383</b>
<b>Index</b> .....	<b>391</b>

## Importing and simulating data

One of the more frustrating tasks associated with the use of any data analysis software concerns the importation of data. Data can be imported into R/qtl in a variety of formats, but users often have trouble with this step. In this chapter, we describe how to import QTL mapping data into R for use with R/qtl. We further discuss the simulation of QTL mapping data. In an optional section, we describe the internal format that R/qtl uses for QTL mapping data.

As this may be the reader's first exposure to R, we will introduce some of the basic aspects of R as we go along. We should again emphasize that the novice user will benefit by spending a couple of days reading Dalgaard (2002) and playing with R.

Before you do anything, you must install R and the R/qtl package; this is described in Appendix A. After invoking R, you must type `library(qtl)` to load the R/qtl package. (In R, R/qtl is known as the `qtl` package or library.) It is best to create a `.Rprofile` file containing this command, so that the package will automatically be loaded whenever you invoke R. (See Sec. A.3.)

Essentially all tasks in R are performed via *functions*, such as the `library` function mentioned above. Appendix B contains partial list of the functions in R/qtl. A complete list may be viewed by typing the following.

```
> library(help=qtl)
```

The `>` symbol is the R prompt, which you will observe when R is ready to accept input commands. R commands may be spread over several lines, in which case the R prompt turns into the `+` symbol, indicating a continuation line. (Appearance of the `+` prompt when one believes one's command is complete may indicate imbalance in parentheses. Press the escape key to cancel the command.) R input will be shown in a *slanted typewriter font*, while output will be in a *plain typewriter font*. (The output for the above command was suppressed, as it would fill a couple of pages.)

Note that the up and down arrow keys may be used to scroll back through previously entered commands. Emacs users will be pleased to find that many

of the Emacs key bindings may be used. (But be careful about Ctrl-p, which may lead you to print a page.)

## 2.1 Importing data

Importing QTL mapping data into R is accomplished with the `read.cross` function. Data may be read in a variety of formats. We strongly recommend the comma-delimited formats discussed in the next subsection, but Map-Maker/QTL, Map Manager QTX, and QTL Cartographer formats may also be used. Sample data files in most of the formats are available at the R/qrtl web site (<http://www.rqtl.org/sampledata>). The help file for `read.cross` contains the complete details on the file formats and the use of the function. The help file may be viewed by typing `?read.cross`; see Sec. A.5. Note that basic use of the `read.cross` function is described in Sec. 2.1.1 on the comma-delimited formats and is not repeated in the subsections on the other formats.

Before contemplating loading one's data into R, it must be assembled into one of the accepted formats. While the comma-delimited formats can be created with Microsoft Excel or other spreadsheet programs, a different format (or computer program) might be best for entering the data into the computer. (And ideally data should enter the computer directly from the measurement device, rather than be input by hand.) The reformatting of data files to conform to the requirements of specific software is a frequent task for geneticists, and hand manipulation of data files is time-consuming and error-prone. Thus we recommend that geneticists learn to program in a language like Perl, which will greatly simplify the task. While the up-front investment to learn Perl is large, the value such knowledge will provide over one's career is far larger.

### 2.1.1 Comma-delimited files

The recommended format for QTL mapping data to be imported into R/qrtl is the comma-delimited format, "`csv`" (an abbreviation of "comma-separated values"). Several variations on this format will be described below. We begin by discussing the basic one.

In the basic "`csv`" format, all phenotype and genotype data, plus the genetic map of the typed markers, are combined into a single file with fields delimited by commas. The file may be constructed in a spreadsheet, such as Microsoft Excel; an example is illustrated in Fig. 2.1. Be careful about the use of commas within the fields (though the use of quotation marks should prevent this from being a problem).

The initial columns are phenotypes (at least one phenotype must be included, such as a numeric index for each individual). Subsequent columns are markers. The first row contains the phenotype and marker names. The second

	A	B	C	D	E	F	G	H	I	J
1	pheno	sex	pgm	c1m1	c1m3	c1m4	c1m5	c2m1	c2m2	c2m3
2				1	1	1	1	2	2	2
3				8.3	49.0	59.5	89.0	1.0	15.0	45.0
4	0.093	f	0	B	B	-	H	B	B	B
5	0.177	f	0	H	H	H	H	H	H	H
6	-	f	0	H	B	A	A	H	-	B
7	0.230	f	0	B	H	H	-	A	A	A
8	0.228	f	0	B	-	H	H	H	B	B
9	0.279	f	0	B	B	A	H	A	H	H
10	0.419	f	0	H	H	H	H	A	B	B
11	0.427	f	0	-	A	B	B	B	H	H
12	0.282	f	0	-	A	B	B	A	A	A
13	0.400	f	0	H	B	A	A	-	H	H
14	0.521	f	0	B	H	B	B	H	H	H
15	0.385	f	0	H	B	A	A	B	H	-
16	0.518	f	0	-	H	H	H	H	H	B

**Figure 2.1.** Part of a data file in the "csv" format, as it might be viewed in a spreadsheet.

row must have empty fields in each of the phenotype columns. (This is quite rigid; even a space character will mess things up.) For the genotype columns, the second row should contain chromosome assignments. Numbers are best; character strings, such as "Chr 1" or "six" will make later data manipulation more cumbersome. Use "X" or "x" to identify the X chromosome.

An optional third row can contain the centiMorgan (cM) positions of the genetic markers. The fields in the phenotype columns should again be blank. Marker order is taken from the cM positions, if provided; otherwise it is taken from the column order.

Subsequent rows correspond to the individuals, with phenotypes followed by genotypes. Missing data should be indicated by "NA" or "-" or some other code. (It is always best to insert some code indicating missingness rather than leave some cells empty, as empty cells can be ambiguous: was the value missing or was an error in data entry made?) Multiple missing data codes may be used, but consistency between the phenotype and genotype data is required: a missing value code for the genotype data cannot be a legitimate phenotype and vice versa. No missing values are allowed in the chromosome identifiers or genetic map positions.

For a backcross, two genotype codes are to be used: one for homozygotes (e.g., AA) and one for heterozygotes (AB). For an intercross, five genotype codes may be used: the two homozygotes (AA and BB), the heterozygote (AB), and two further genotype codes to be used for dominant markers, such as D for "not BB" (i.e., AA or AB) and C for "not AA" (i.e., AB or BB), as used by the MapMaker software.

Consistency in genotype codes is required: one cannot use both A and AA to indicate a homozygous A genotype. Also note that spaces can mess things

**Table 2.1.** Possible intercrosses, and the appropriate code for the `pgm` “phenotype.” In the crosses, females are always listed first, so  $A \times B$  means a female A crossed to a male B.

Cross	Possible genotypes		pgm code
	Females	Males	
$(A \times B) \times (A \times B)$	AA, AB	A-, B-	0
$(B \times A) \times (A \times B)$	AA, AB	A-, B-	0
$(A \times B) \times (B \times A)$	AB, BB	A-, B-	1
$(B \times A) \times (B \times A)$	AB, BB	A-, B-	1

up: “A ” is treated as different from “A”. It is best to ensure that there are no spaces in the final data file.

X chromosome genotypes should be coded just like the autosomal genotype data; in particular, hemizygous males should be coded as if they were homozygous, rather than using separate codes for hemizygous and homozygous genotypes. If X chromosome genotype data are included, one of the phenotypes should indicate the sex of the individuals. This may be called “sex” or “Sex,” and the sexes may be coded by 0/1 for females/males, or by the codes `f/m`, `F/M`, or `female/male`.

Further care is required for the X chromosome genotype data in an intercross, as the direction of the cross must be known. Four possible intercrosses may be performed, as shown in Table 2.1. In all cases, the males are hemizygous A or B at any one locus, but in the crosses  $(A \times B) \times (A \times B)$  and  $(B \times A) \times (A \times B)$ , the females are either AA or AB, while in the crosses  $(A \times B) \times (B \times A)$  and  $(B \times A) \times (B \times A)$ , the females are either AB or BB. Thus, the order of the cross producing the  $F_1$  male is critical; for example, we wish to know whether the paternal grandmother was from strain A or B.

We thus require, for intercrosses, a “phenotype” column named `pgm` (for “paternal grandmother”), with codes 0 and 1 indicating which individuals came from which cross, as shown in Table 2.1.

If one includes a phenotype named “`id`,” “`ID`,” or “`Id`,” it will be assumed to provide individual identifiers. These will be used in certain places to indicate the individuals (such as in `plot.geno`; see Sec. 3.5).

The specification of a file in the “`csv`” format is now complete. If the file was created in a spreadsheet program, such as Microsoft Excel, you will need to use “Save as” and select the format “CSV (comma-delimited)” to create the actual file. The result will look something like that shown in Fig. 2.2. A complete example is provided at the R/qtl web site.

With our first file format understood, we now turn to the use of `read.cross` to load the data into R.

A list of the input arguments for `read.cross` may be viewed via the `args` function, as follows. (We often use `args` to get a quick reminder of the input to a function.) Remember that, if R/qtl is not yet loaded, one must use

```

pheno,sex,pgm,c1m1,c1m3,c1m4,c1m5,c2m1,c2m2,c2m3,c2m4,...
,,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,5,5,5,5...
,,8.3,49,59.5,89,1,15,45,68.9,80.9,87.4,99,0,11.2,39...
0.093,f,0,A,B,A,A,B,H,H,H,H,H,B,H,H,B,B,H,A,A,A,A,H,...
0.177,f,0,B,H,H,H,H,B,B,B,B,H,B,H,H,H,-,H,H,H,H,A,H,A,...
0.271,f,0,B,A,H,H,H,H,H,A,A,A,A,H,H,H,H,H,H,B,-,B,B,H,...
0.230,f,0,B,B,A,H,B,B,B,B,B,B,B,A,H,H,B,B,H,H,H,B,B,H,...
0.228,f,0,H,H,H,H,H,H,H,H,H,B,B,B,B,H,H,H,H,B,H,H,H,B,...
0.279,f,0,H,B,A,A,H,B,B,H,A,A,-,A,A,A,H,H,H,A,A,H,B,H,...
0.419,f,0,B,H,H,H,A,A,A,A,A,A,B,B,B,B,B,B,H,H,H,H,...
0.427,f,0,B,A,H,H,H,B,B,B,B,B,H,H,H,B,B,B,H,H,A,A,A,...
0.282,f,0,B,B,A,H,A,H,H,A,A,A,A,B,B,H,A,-,B,H,H,H,H,B,...
0.4,f,0,H,H,H,H,A,B,B,H,H,H,H,H,H,B,H,H,H,H,H,H,H,...
0.521,f,0,H,A,B,B,B,H,H,H,H,H,H,H,A,A,A,H,B,B,B,H,H,...
0.385,f,0,A,A,B,B,A,A,A,H,H,H,H,B,B,H,H,H,H,H,A,A,A,...
0.518,f,0,H,B,A,A,H,H,H,B,B,B,B,A,A,H,H,A,H,H,H,H,H,H,...
:

```

**Figure 2.2.** Part of a text file in the "csv" format. The terminal dots in each line are just to indicate that the file extends quite far to the right.

the `library` function to make it available. (Ignore the `NULL`; that's just a meaningless bit from the `args` function.)

```

> library(qtl)
> args(read.cross)

function (format = c("csv", "csvr", "csvs", "csvsr", "mm",
  "qtx", "qtlcart", "gary", "karl"), dir = "", file, genfile,
  mapfile, phefile, chrfile, mnamesfile, pnamesfile,
  na.strings = c("-", "NA"), genotypes = c("A", "H", "B", "D",
  "C"), alleles = c("A", "B"), estimate.map = TRUE,
  convertXdata = TRUE, ...)
NULL

```

This is, admittedly, rather forbidding, but not all of the arguments will be needed in all cases. Note that the `c` function is used to combine multiple items together into a vector.

The argument `format` will be used to indicate that we are reading data in the "csv" format. The possible formats are shown; the first listed is taken as the default. The argument `dir` is used to indicate the directory in which the file appears. By default, it is assumed that the file is in the current working directory. (For details on how to select or change the working directory, see

Sec. A.4.) The argument `file` will be used to give the name of the data file. The other file arguments are used for formats in which the data are split across multiple files.

The argument `na.strings` is used to indicate the set of missing data codes. By default, either “-” or “NA” will be treated as missing. Note that most things are case-sensitive, so “na” will be treated as different from “NA” and “Na”. If all of these appear in the data file, all should be indicated via the `na.strings` argument.

The argument `genotypes` is used to indicate the genotype codes, and takes a vector of character strings. The order of the codes in the string is important. We often forget whether “D” stands for “not AA” or “not BB,” and so we generally must refer to the help file for `read.cross`, where this is explained. Note, again, that the codes are case-sensitive, so “a” will be treated as different from “A.”

The argument `alleles` is used to indicate custom names for the alleles (single-character names are best), so that if one does a mouse cross of BALB/c  $\times$  DBA/2, one might want to use the codes C and D for the alleles. These will be used in certain plots (such as of phenotype against genotype) and summaries.

If the genetic map positions of the markers are not provided in the file and `estimate.map=TRUE`, the intermarker distances will be estimated, while if `estimate.map=FALSE`, a dummy map will be created. (If genetic map positions *are* provided, this argument will be ignored.) Estimation of the genetic map can sometimes be time-consuming, and so one may wish to use `estimate.map=FALSE`. One may later estimate the map with the function `est.map` and plug it into the data object with `replace.map`; see Sec. 3.4.3.

If marker positions are provided in the file, it is important that no two markers are placed at precisely the same position. If they are, this may be rectified with the function `jittermap`; see page 84.

The “...” at the end of the specification of `read.cross` is used to allow additional arguments to be specified; these are passed to the more basic R function `read.table`, which does the actual work of reading in the data. Its use will be explained further below.

There seems a lot to understand, but use of `read.cross` is generally not so tedious as it might appear, as most of the arguments to the function can be ignored. For example, suppose the data in Figures 2.1 and 2.2 are saved in one’s working directory as the file `mydata.csv`. One could read this into R with the following.

```
> mydata <- read.cross("csv", "", "mydata.csv")
```

Note that “<-” is the *assignment operator*. The data are read from the `mydata.csv` file and combined into a single object (with a very special internal format, described in Sec. 2.6.1) and assigned to `mydata`. This will be a new object in our R workspace that we may manipulate and analyze. Type `ls()` or `objects()` to list the objects in your workspace.

Also note that arguments to functions in R may be specified by their position in the list, by their name, or they may be left unspecified (in which case

the default values are assumed). Thus, in the code above, it is assumed that `format="csv"`, `dir=""`, and `file="mydata.csv"`, and we need not specify values for `na.strings`, `genotypes`, or `alleles`, as the default values suffice for our data. All of the following lines of code are equivalent.

```
> mydata <- read.cross("csv", , "mydata.csv")
> mydata <- read.cross("csv", file="mydata.csv")
> mydata <- read.cross(format="csv", file="mydata.csv")
> mydata <- read.cross(file="mydata.csv", format="csv")
> mydata <- read.cross(file="mydata.csv")
```

If the data file were in some location other than the R working directory, we would need to specify its location with the `dir` argument. The directory (or folder) hierarchy is indicated with forward slashes (/). In Windows, it is traditional to use backslashes (\), but these will not work in R, though double-backslashes (\\) may be used in place of forward slashes.

For example, if we were working on a Macintosh and our file was on the Desktop, we might use the following code. The tilde (~) denotes our home directory.

```
> mydata <- read.cross("csv", "~/Desktop", "mydata.csv")
```

If we were working in Windows and the file was located in `c:\My Data`, we could use the following code.

```
> mydata <- read.cross("csv", "c:/My Data", "mydata.csv")
```

If we had coded the genotype data differently, we would need to use the `genotypes` argument. Because of all of the intervening file name arguments, the `na.strings`, `genotypes`, and `alleles` arguments generally must be specified by name. For example, suppose missing data were coded “na” and that the genotypes were coded BB/BC/CC. Then the data would be read as follows.

```
> mydata <- read.cross("csv", "", "mydata.csv", na.strings="na",
+                       genotypes=c("BB", "BC", "CC"),
+                       alleles=c("B", "C"))
```

We recommend downloading the example “csv” data file (`listeria.csv`) from the R/qtl web site and trying to load it into R. (The file is included with the R/qtl package, but it is in a spot that may be difficult to find.) If one has trouble importing one’s own data, it is a good idea to try importing a file that is known to be correct, so one may determine whether the problem concerns some incompatibility in the file or an incomplete understanding of the use of `read.cross`.

Outside the United States, commas are sometimes used instead of periods in numbers, and so semicolons are sometimes used instead of commas in such CSV files. Files of this sort may also be read; one must make use of the flexibility in the `read.cross` function through the “...” in its specification, through



```

# The ch3c data
# File created by Karl W Broman, 7-19-06
# Intercross between C57BL/6J and A/J
# 100 females from the cross (AxB)x(AxB)
# 101 markers, including 10 on the X chromosome
pheno,sex,pgm,c1m1,c1m3,c1m4,c1m5,c2m1,c2m2,c2m3,c2m4,...
,,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,5,5,5,5,...
,,8.3,49,59.5,89,1,15,45,68.9,80.9,87.4,99,0,11.2,39....
0.093,f,0,A,B,A,A,B,H,H,H,H,H,B,H,H,B,B,H,A,A,A,A,H,...
0.177,f,0,B,H,H,H,H,B,B,B,B,H,B,H,H,H,-,H,H,H,H,A,H,A,...
0.271,f,0,B,A,H,H,H,H,H,A,A,A,A,H,H,H,H,H,B,-,B,B,H,...
0.230,f,0,B,B,A,H,B,B,B,B,B,H,B,A,H,H,B,B,H,H,H,B,B,H,...
0.228,f,0,H,H,H,H,H,H,H,H,B,B,B,B,H,H,H,H,B,H,H,H,B,...
0.279,f,0,H,B,A,A,H,B,B,H,A,A,-,A,A,A,H,H,H,A,A,H,B,H,...
0.419,f,0,B,H,H,H,A,A,A,A,A,A,B,B,B,B,B,B,H,H,H,H,...
0.427,f,0,B,A,H,H,H,B,B,B,B,B,H,H,H,B,B,B,H,H,A,A,A,...
:

```

**Figure 2.3.** An example file in the "csv" format with comment lines included.

which further arguments are passed down to the more basic `read.table` function. That function allows arguments `sep`, for specifying the field separator, and `dec`, for specifying the character used for the decimal point.

Thus, if the `mydata.csv` file had used semicolons and commas rather than commas and periods, we would read it into R with the following code.

```
> mydata <- read.cross("csv", , "mydata.csv", sep=";", dec=",")
```

Note that these additional arguments *must* be specified by name.

One may include comments in an input file, to be ignored when it is imported, but useful to document its contents. A single symbol, such as `#`, may be used to indicate that the remainder of the line is to be ignored. The chosen symbol cannot appear anywhere in the data, and is indicated, in the call to `read.cross`, via the `comment.char` argument. (In R versions 2.3.1 and earlier, `comment.char="#"` was the default, but in R versions 2.4.0 and later, the default has become `comment.char=""`, and so no such commenting character is assumed.)

For example, the file in Fig. 2.3 contains initial comment lines, indicated by `#`. To read this file into R, we would use the following code.

```
> mydata <- read.cross("csv", , "mydata.csv", comment.char="#")
```

There are three related comma-delimited formats: "csvr", "csvs", and "csvsr". These are primarily for the case of expression genetic data, in which

	A	B	C	D	E	F	G	H	I	J
1	pheno			0.093	0.177	-	0.230	0.228	0.279	0.419
2	sex			f	f	f	f	f	f	f
3	pgm			0	0	0	0	0	0	0
4	c1m1	1	8.3	B	H	H	B	B	B	H
5	c1m3	1	49.0	B	H	B	H	-	B	H
6	c1m4	1	59.5	-	H	A	H	H	A	H
7	c1m5	1	89.0	H	H	A	-	H	H	H
8	c2m1	2	1.0	B	H	H	A	H	A	A
9	c2m2	2	15.0	B	H	-	A	B	H	B
10	c2m3	2	45.0	B	H	B	A	B	H	B
11	c2m4	2	68.9	B	H	H	A	B	A	H
12	c2m5	2	80.9	B	B	A	A	B	A	H
13	c2m6	2	87.4	H	B	A	A	B	A	H
14	c2m7	2	99.0	B	B		-	B	A	H
15	c3m1	3	0.0	A	B	A	B	H	B	H
16	c3m2	3	11.2	H	B	-	B	H	B	H

**Figure 2.4.** Part of a data file in the "csvr" format, as it might be viewed in a spreadsheet.

QTL mapping is to be performed with the expression of all genes on a microarray, so that one has thousands or tens of thousands of phenotypes.

The "csvr" format is just like the "csv" format, but with rows and columns interchanged. (The "r" is for *rotate*, but the file is technically *transposed* rather than rotated.) In Fig. 2.4, the file from Fig. 2.1 is shown in the "csvr" format. All other aspects are the same as before, and the use of `read.cross` is unchanged, so such a file (call it "mydata\_rot.csv") could be read in as follows.

```
> mydata <- read.cross("csvr", , "mydata_rot.csv")
```

Of course, other arguments, such as `genotypes`, may be used as before.

The "csvs" format is similar to the "csv" format, but with separate files for the phenotypes and the genotypes. The genotype data file must begin with a single column containing individual identifiers, followed by columns for each of the markers. As with the phenotype columns for the "csv" format, this initial column must have empty cells in the rows for the chromosome assignments and marker positions. The phenotype data file must contain a column with precisely the same name and contents, so that we can be sure that the phenotype and genotype data are appropriately aligned. An example of this format is display in Fig. 2.5.

To read data in the "csvs" format, one must specify the names of both files. This may be done via the `read.cross` arguments `genfile` and `phefile`, as follows. (We assume that both files are in the current working directory.)

```
> mydata <- read.cross("csvs", genfile="mydata_gen.csv",
+                       phefile="mydata_phe.csv")
```

Phenotype data file					Genotype data file					
	A	B	C	D	A	B	C	D	E	
1	pheno	sex	pgm	id	1	id	c1m1	c1m3	c1m1	c1m3
2	0.093	f	0	1	2		1	1	1	1
3	0.177	f	0	2	3		8.3	49.0	8.3	49.0
4	-	f	0	3	4	1	B	B	B	B
5	0.230	f	0	4	5	2	H	H	H	H
6	0.228	f	0	5	6	3	H	B	H	B
7	0.279	f	0	6	7	4	B	H	B	H
8	0.419	f	0	7	8	5	B	-	B	-
9	0.427	f	0	8	9	6	B	B	B	B
10	0.282	f	0	9	10	7	H	H	H	H
11	0.400	f	0	10	11	8	-	A	-	A
12	0.521	f	0	11	12	9	-	A	-	A
13	0.385	f	0	12	13	10	H	B	H	B

**Figure 2.5.** Part of the genotype and phenotype data files for an example of the "csvs" format, as they might be viewed in a spreadsheet.

For the user's convenience, if the `phedata` argument was not specified, but the `file` and `genfile` arguments were, we assume that `file` and `genfile` are indicating the genotype and phenotype data files, respectively. This can simplify the code a bit. For example, suppose that we are working in a directory `MyProject/R`, and that the two data files are sitting in the directory `MyProject/Data`. The data could be imported as follows.

```
> mydata <- read.cross("csvs", "../Data", "mydata_gen.csv",
+                      "mydata_phe.csv")
```

The "csvsr" format is just like the "csvs" format, but with both files rotated as in the "csvr" format. We use `read.cross` in the same way as for the "csvs" format.

### 2.1.2 MapMaker/QTL

The format "mm" is for data in the format used by the MapMaker software. There are two files, a `.raw` file containing the genotype and phenotype data and a second file containing the genetic map information. Examples of these files are provided on the R/qtl web site.

The genetic map file may be in one of two formats. First, one may use a `.maps` file, produced by MapMaker/Exp. Second, one may create a space-delimited file, as illustrated in Fig. 2.6, with one row for each marker. The first column is the chromosome assignment, the second column is the marker name (which must match that used in the `.raw` file exactly), and an optional third column may contain the cM position of each marker.

Use of `read.cross` to read data in the "mm" format is similar to the case of the "csvs" format, discussed in the previous subsection. Specify the `.raw` file

```

1 D10M44 0.00
1 D1M3 1.00
1 D1M75 24.85
1 D1M215 40.41
1 D1M309 49.99
1 D1M218 52.80
1 D1M451 70.11
1 D1M504 70.81
1 D1M113 80.62
1 D1M355 81.40
1 D1M291 84.93
1 D1M209 92.68
1 D1M155 93.64
2 D2M365 0.00
2 D2M37 27.94
2 D2M396 47.11
:

```

**Figure 2.6.** The initial portion of a space-delimited file that may be used to indicate marker locations for the MapMaker ("mm") format.

with the `file` argument and the genetic map file with the `mapfile` argument. (The format of the genetic map file is determined automatically.) Note that the `na.strings` and `genotypes` arguments are ignored with this format, as such codes are specified within the `.raw` file.

For the user's convenience, if the `mapfile` argument was not specified, but the `genfile` argument was, we assume that `genfile` indicates the genetic map file. This can simplify the code a bit. For example, suppose that we are working in a directory `MyProject/R`, and that the two data files are sitting in the directory `MyProject/Data`. The data could be imported as follows.

```

> mydata <- read.cross("mm", "../Data", "mydata.raw",
+                      "mydata.maps")

```

### 2.1.3 QTL Cartographer

The format "qtlcart" is for data in the format used by the QTL Cartographer software. There are two files, a `.cro` file containing the genotype and phenotype data and a `.map` file containing the genetic map. Examples of these files are provided on the R/qtl web site.

We use `read.cross` to read the QTL Cartographer files in a manner similar to that used for the MapMaker files. For example, suppose we are working in

a directory `MyProject/R` and that the two data files are in the directory `MyProject/Data`; they could then be imported as follows.

```
> mydata <- read.cross("qtlcart", "../Data", "mydata.cro",
+                      "mydata.map")
```

### 2.1.4 Map Manager QTX

The format "qtx" is for data in the format used by the Map Manager QTX software. There is a single file, generally with extension `.qtx`, containing all of the genotype and phenotype data as well as the genetic markers' chromosome assignments and order. Genetic map positions for the markers are generally not included in the file, and so must be estimated. An example file is provided on the R/qtl web site.

Loading data from a `.qtx` file into R/qtl is simple. The `na.strings` and `genotypes` arguments need not be used, as such codes are included within the file. Suppose that we are working in the directory `MyProject/R`; to read the `mydata.qtx` from the directory `MyProject/Data`, type the following.

```
> mydata <- read.cross("qtx", "../Data", "mydata.qtx")
```

As the genetic map positions for the markers are generally not provided in the `.qtx` file, and so must be estimated from the data, the import of the data can be time consuming. One may wish to use `estimate.map=FALSE` in the call to `read.cross`, and then use `est.map` and `replace.map` to estimate the map and then plug it into the data. This process is described in more detail in Sec. 3.4.3, but let us briefly consider a simple example.

```
> mydata <- read.cross("qtx", "", "mydata.qtx",
+                      estimate.map=FALSE)
> themap <- est.map(mydata, error.prob=0.001)
> mydata <- replace.map(mydata, themap)
```

In the first line of code, we read in the data without estimating the intermarker distances, and so a dummy map is inserted into the `mydata` object. In the second line, we call `est.map` to estimate the genetic map, here assuming that genotypes may be in error with probability 0.1%. The result is placed in the object `themap`. In the final line of code, the `replace.map` function is used to replace the map within `mydata`, inserting `themap` in its place. The output is the same data but with a different map; we assign it back to `mydata`, writing over the original data. (We might have assigned it to an object with a different name, in which case both would appear in our R workspace.)

## 2.2 Exporting data

Data may be exported from R/qtl into several formats. This may be useful, for example, if one wishes to compare results from R/qtl to those from QTL Cartographer, or simulate data in R/qtl and analyze them in Cartographer.

The `write.cross` function is used for this purpose. The `cross` argument is the cross object to be exported. The `chr` argument may be used to indicate a subset of chromosomes that should be exported. The `format` argument indicates the format to which the data should be written.

The `filestem` argument indicates the initial part of the file names. For example, with the `qtlcart` format, `.cro` and `.map` files will be created. If one uses `filestem="mydata"`, the files `"mydata.cro"` and `"mydata.map"` will be created.

The `filestem` can include a directory, so that the files may be written somewhere other than the current working directory. For example, if one wishes to save chromosomes 5 and 13 of the `listeria` data to a file in the `"csv"` format on the Desktop on a Macintosh computer, use the following code.

```
> data(listeria)
> write.cross(listeria, "csv", "~/Desktop/listeria", c(5, 13))
```

## 2.3 Example data

A variety of example data sets are included with R/qtl. A complete list may be obtained with the following.

```
> data(package="qtl")
```

Of particular interest are the `hyper` and `listeria` data, which will be used as the main examples in this book.

The `hyper` data set is from Sugiyama *et al.* (2001). (It was also discussed in Sec. 1.2.) This is a backcross using the C57BL/6J and A/J inbred mouse strains, with the F<sub>1</sub> mated back to the C57BL/6J strain. There are 250 male backcross individuals. Mice were given water containing 1% NaCl for two weeks; the phenotype is blood pressure (actually the average of 20 blood pressure measurements from each of 5 days).

The `listeria` data set is from Boyartchuk *et al.* (2001). This is an intercross using the C57BL/6ByJ and BALB/cByJ inbred mouse strains. There are 120 female intercross individuals (though only 116 were phenotyped). Mice were injected with *Listeria monocytogenes*; the phenotype is survival time (in hours). A large proportion of the mice (35/116) survived past the 240-hour time point and were considered to have recovered from the infection; their phenotype was recorded as 264.

A number of further example data sets will be used in this book. (For a summary of all data sets considered in the book, see Appendix C.) These have been compiled into an R package, R/qtlbook (known in R as the `qtlbook` package). It may be obtained from the website for the book (<http://www.rqtl.org/book>) and from the Comprehensive R Archive Network (CRAN, <http://cran.r-project.org>).

Additional example data may be obtained at the QTL Archive at The Jackson Laboratory (<http://cgd.jax.org/nav/qtlarchive1.htm>). Most of the data sets are available in the "csv" format. One must register to access the data. As stated at the QTL Archive, "The authors of the datasets retain individual ownership of the data. We request, as a courtesy to the authors, that you alert them in advance of any publications that result from reanalysis of these data or obtain permission prior to redistribution of data or results."

## 2.4 Data summaries

All of the data read by `read.cross` (including genotypes, phenotypes, and the genetic map) will be stored in a single object. (This object is stored in a quite complex form; see Sec. 2.6.1.) A number of functions are provided to get summary information about the object.

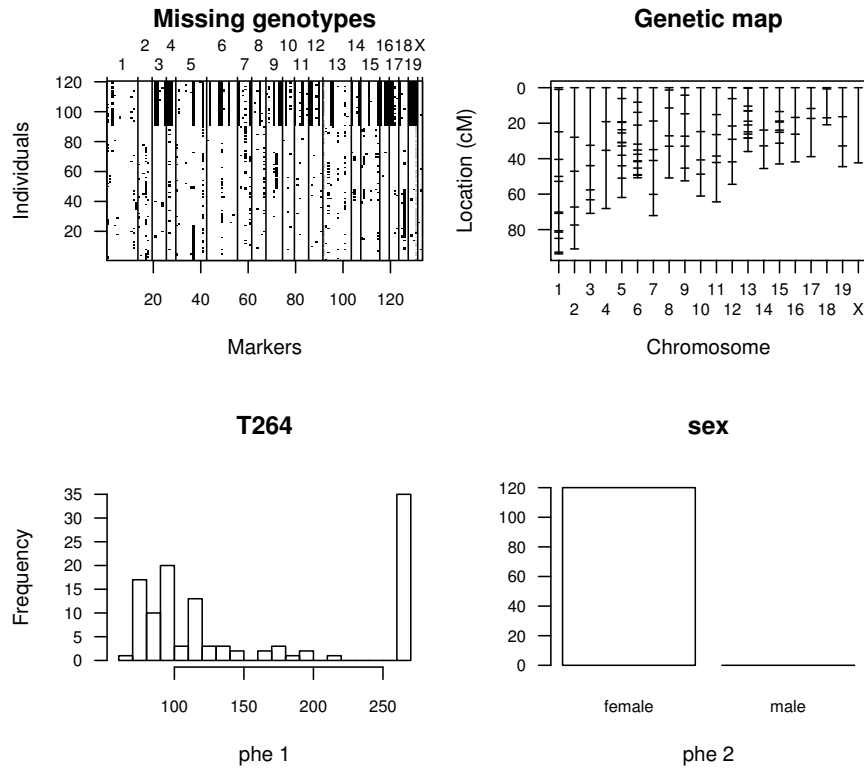
The most important function is `summary.cross`. In addition to providing a brief summary of the cross, it performs an extensive series of checks of the integrity of the data (for example, that there are the same number of individuals in the phenotype data as in the genotype data).

The data object for a QTL mapping experiment is assigned a "class" "`cross`". R includes some simple object-oriented features, so that one may use the generic functions `summary` and `plot` on an object, and the relevant summary or plot is made.

For example, the following code loads the `listeria` data and displays a brief summary.

```
> data(listeria)
> summary(listeria)
```

F2 intercross	
No. individuals:	120
No. phenotypes:	2
Percent phenotyped:	96.7 100
No. chromosomes:	20
Autosomes:	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
X chr:	X
Total markers:	133
No. markers:	13 6 6 4 13 13 6 6 7 5 6 6 12 4 8 4 4 4 4 2
Percent genotyped:	88.5



**Figure 2.7.** The summary plot of the `listeria` data provided by the `plot.cross` function, including the pattern of missing genotype data (upper left; black pixels indicate missing data), the genetic map of the typed markers (upper right), a histogram of the phenotype (lower left), and a bar plot of the sexes (lower right).

```
Genotypes (%):   CC:26.2  CB:48.9  BB:24  not BB:0
                  not CC:0.9
```

We see that this is an intercross with 120 individuals, that there are two phenotypes, and 20 chromosomes containing 133 markers, and with genotype completion of 88.5%.

In the above code, the generic `summary` function sees that `listeria` has class "cross" and passes it to the `summary.cross` function, which provides the actual summary.

Similarly, the following code provides a summary plot of the `listeria` data, and in this case the generic `plot` function passes `listeria` to the `plot.cross` function, which makes the plot (shown in Fig. 2.7).

```
> plot(listeria)
```

The individual panels in Fig. 2.7 may be obtained with the following code.



```

> plot.missing(listeria)
> plot.map(listeria)
> plot.pheno(listeria, 1)
> plot.pheno(listeria, 2)

```

The `plot.missing` function creates the plot with the pattern of missing genotype data. It takes an argument `reorder` which can be used to order the individuals according to their phenotype. The genetic map is obtained with `plot.map`. The function `plot.pheno` plots a phenotype, either as a histogram (using the R function `hist`) or as a bar plot (using the R function `barplot`), depending on the nature of the phenotype.

Finally, there are a variety of other functions for getting additional small pieces of information about a cross object. They are largely self-explanatory.

```

> nind(listeria)

[1] 120

> nphe(listeria)

[1] 2

> totmar(listeria)

[1] 133

> nchr(listeria)

[1] 20

> nmar(listeria)

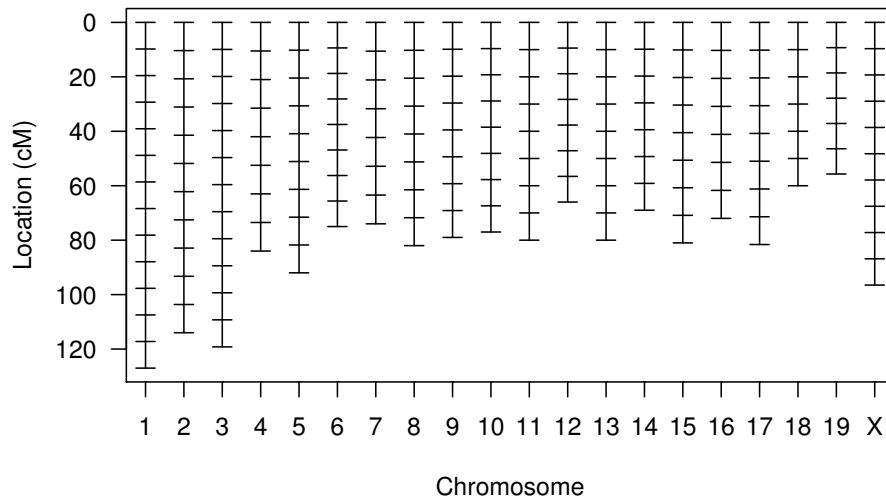
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19  X
13  6  6  4 13 13  6  6  7  5  6  6 12  4  8  4  4  4  4  2

```

The function `nmar` gives the numbers of markers on individual chromosomes.

## 2.5 Simulating data

One can simulate QTL mapping data in R/qtl with the `sim.cross` function; it can simulate only additive QTL models. These basic facilities are described in the next subsection. More complex QTL models may also be simulated by making use of the QTL genotype data, which are stored in the object output by `sim.cross`. This will be described in the following subsection. Computer simulations are particularly useful for exploring the power to detect QTL and the precision of localization of QTL. For further details, see Sec. 6.6.



**Figure 2.8.** A genetic map, with approximately 10 cM marker spacing, modeled after the mouse genome and contained in the `map10` data set in `R/qt1`.

### 2.5.1 Additive models

The `sim.cross` function may be used to simulate a backcross or intercross with an additive QTL model. It requires, as input, a genetic map of markers. Such a map must be stored in a specific and rather complicated form (see Sec. 2.6.2), and so we first describe how to create a map.

First, an example map, modeled after the mouse genome and having approximately evenly spaced markers (at  $\sim 10$  cM) is provided with `R/qt1` in the data set `map10`. To access the object and plot the map, type the following.

```
> data(map10)
> plot(map10)
```

The plot is shown in Fig. 2.8. The marker spacing varies slightly across chromosomes so that the lengths of the chromosomes match those of the mouse genome.

Second, one may extract the genetic map from a QTL mapping data set with the `pull.map` function. For example, the following code extracts the map from the `listeria` data.

```
> data(listeria)
> listmap <- pull.map(listeria)
```

Finally, one may use `sim.map` to generate a map, with equally spaced markers or with markers placed randomly. Important arguments to `sim.map` include `len` (the cM lengths of the chromosomes), `n.markers` (the numbers of markers on the chromosomes), `anchor.tel` (indicates whether the ends of the chromosomes should be forced to have markers), `include.x` (whether the

final chromosome should be designated to be the X chromosome, versus all chromosomes being autosomes), and `eq.spacing` (whether markers should be spaced evenly).

For example, to create a map with a single autosome of length 200 cM and having markers equally spaced at 20 cM, type the following.

```
> mapA <- sim.map(200, 11, include.x=FALSE, eq.spacing=TRUE)
```

To create a map with 19 autosomes and an X chromosome, chromosomes all of length 100 cM, and each containing 10 randomly positioned markers, though ensuring one marker at each end of each chromosome, we would type the following.

```
> mapB <- sim.map(rep(100, 20), 10)
```

A similar map, but without anchoring the telomeres, would be obtained as follows.

```
> mapC <- sim.map(rep(100, 20), 10, anchor.tel=FALSE)
```

Finally, to get a map with four autosomes of lengths 50, 75, 100, and 125 cM, respectively, and with equally spaced markers at a 5 cM spacing, type the following.

```
> L <- c(50, 75, 100, 125)
> mapD <- sim.map(L, L/5+1, eq.spacing=TRUE, include.x=FALSE)
```

Note that one can use the `summary.map` function to get a short summary of a genetic map; it works much like the `summary.cross` function described in Sec. 2.4. We can get a summary of the `mapD` object, created above, as follows.

```
> summary(mapD)
```

	n.mar	length	ave.spacing	max.spacing
1	11	50	5	5
2	16	75	5	5
3	21	100	5	5
4	26	125	5	5
overall	74	350	5	5

With a genetic map in hand, we can now turn to the simulation of the actual data. The following code simulates data for a backcross of 100 individuals, with complete and error-free genotype data, and markers placed according to the genetic map in `map10`.

```
> simA <- sim.cross(map10, n.ind=100, type="bc")
```

We would simulate an intercross in the same way, using `type="f2"`.

If QTL are to be simulated, we must specify the model via the `model` argument, which should be a matrix with three columns for a backcross and four columns for an intercross. The first column in the matrix gives the chromosomes on which the QTL sit and the second column gives their cM positions.

The third column contains the additive effect of each QTL: in a backcross, the difference between the phenotype averages in heterozygotes and homozygotes; in an intercross, half the difference between phenotype averages for the homozygotes. In an intercross, there must be a fourth column giving the dominance effect for each QTL (the difference between the average phenotype for the heterozygotes and the midpoint between the average phenotypes for the homozygotes).

Phenotypes are simulated from a normal distribution with residual variance  $\sigma^2 = 1$ . Thus, in a backcross, if there is one QTL with additive effect  $a$ , the proportion of the phenotypic variance explained by the QTL (i.e., the heritability due to the QTL) will be  $a^2/4/(a^2/4 + 1)$ . In an intercross with one QTL exhibiting no dominance, the proportion of the phenotypic variance explained is  $a^2/2/(a^2/2 + 1)$ .

Let us first simulate a backcross with two additive QTL, each responsible for 8% of the phenotypic variance. Place the first at 50 cM on chromosome 1 and the second at 65 cM on chromosome 14. We must first find the additive effects that correspond to 8% phenotypic variance. Since the QTL are unlinked and have the same size effect, we need  $(a^2/4)/[2(a^2/4) + 1] = 0.08$ . Solving for  $a$ , we obtain  $a = \sqrt{4 \times 0.08/(1 - 2 \times 0.08)}$ .

```
> a <- 2 * sqrt(0.08 / (1 - 2 * 0.08))
> mymodel <- rbind(c(1, 50, a), c(14, 65, a))
> simB <- sim.cross(map10, type="bc", n.ind=200, model=mymodel)
```

We use the `c` function to combine the chromosome, position and effect of each QTL into a vector, and then `rbind` to combine the two into a matrix (`rbind` makes them rows in the matrix).

As a further example, we simulate an intercross of 250 individuals with three QTL, two having no dominance but with effects in the opposite directions and a third being strictly dominant. Let's have the first two QTL be linked on chromosome 3 at positions 40 cM and 65 cM, and place the third on chromosome 4 at 5 cM. For simplicity, let's set the effects at 0.5.

```
> mymodel2 <- rbind(c(3, 40, 0.5, 0), c(3, 65, -0.5, 0),
+                 c(4, 5, 0.5, 0.5))
> simC <- sim.cross(map10, type="f2", n.ind=250, model=mymodel2)
```

By default, there are no errors in the genotype data. Errors can be included at random via the `error.prob` argument. Genotype data are also, by default, complete. The genotype data can be missing at random with some probability via the `missing.prob` argument. And so we can repeat our backcross simulation with 1% genotyping errors and 5% missing data as follows.

```
> simD <- sim.cross(map10, type="bc", n.ind=200, model=mymodel,
+                 error.prob=0.01, missing.prob=0.05)
```

Random missing genotype data is rather artificial. For more realistic missing data, we can simulate an intercross of the same size as the `listeria` data

and apply the missing data observed in that data set. This is not so simple, due to the complexity of the cross data objects and the need for a loop over chromosomes, and so the following code has little chance of being understood by the novice.

```
> data(listeria)
> listmap <- pull.map(listeria)
> simE <- sim.cross(listmap, type="f2", n.ind=nind(listeria),
+                   model=mymodel2)
> for(i in 1:nchr(simE))
+   simE$geno[[i]]$data[ is.na(listeria$geno[[i]]$data) ] <- NA
```

By default, simulations are performed assuming no crossover interference at meiosis. One may also simulate the crosses under the  $\chi^2$  model or the Stahl model. (See Sec. 2.7 for references.) The  $\chi^2$  model has a single parameter,  $m$ , which is a non-negative integer;  $m = 0$  corresponds to no interference. With  $m > 0$ , it is assumed that, on the four-strand bundle at meiosis, chiasmata and intermediate points are thrown down at random (according to a Poisson process), and that every  $(m + 1)$ st point is a chiasma. No chromatid interference is assumed, so that the particular strands involved in each chiasma are at random, independent between chiasmata. As a result, the crossovers on a random meiotic product may be obtained by “thinning” the chiasmata independently with probability  $1/2$ . (That is, each chiasma has  $1/2$  chance of being a crossover on the random product, with independence between chiasmata.) In the Stahl model, chiasmata arise according to two independent mechanisms, one following a  $\chi^2$  model and the other exhibiting no interference; the observed chiasma locations are the superposition of the two processes. There is one additional parameter,  $p$ , giving the proportion of chiasmata to come from the mechanism exhibiting no interference.

We can simulate under the  $\chi^2$  model and the Stahl model via the arguments `m` and `p` to `sim.cross`. By default, `m=0` (in which case `p` is irrelevant), indicating no crossover interference. The mouse exhibits strong crossover interference with  $m \approx 10$ . We can repeat our previous simulation, but with recombination according to a  $\chi^2(m = 10)$  model as follows.

```
> simF <- sim.cross(map10, type="f2", n.ind=250, model=mymodel2,
+                   m=10)
```

We can simulate from the Stahl model, with  $m = 10$  and  $p = 0.1$ , as follows.

```
> simG <- sim.cross(map10, type="f2", n.ind=250, model=mymodel2,
+                   m=10, p=0.1)
```

### 2.5.2 More complex models

The simulations in the previous section were restricted to strictly additive QTL models and with residual variation following a normal distribution with

variance  $\sigma^2 = 1$ . However, the QTL genotype data are stored as a matrix within the output of `sim.cross`; with these data one may simulate data from essentially any QTL model.

First, let us simulate two QTL exhibiting epistasis. Consider a backcross of 200 individuals, with a QTL located at 25 cM on chromosome 4 and another at 45 cM on chromosome 5. Assume that an effect is seen only if an individual is homozygous at both QTL, in which case the phenotype is reduced by one unit.

We begin by simulating QTL having no effect, just so that their genotypes may be obtained, but so that the simulated phenotype will follow a normal(0,1) distribution, independent of genotype. We then modify the phenotype for individuals who are homozygous at both QTL. This requires a bit of mucking about in the cross data object.

```
> data(map10)
> nullmodel <- rbind(c(4, 25, 0), c(5, 45, 0))
> episim <- sim.cross(map10, type="bc", n.ind=200,
+                   model=nullmodel)
> qtlg <- episim$qtlgeno
> wh <- qtlg[,1]==1 & qtlg[,2]==1
> episim$pheno[wh, 1] <- episim$pheno[wh, 1] - 1
```

In the fifth line, we pull out the QTL genotype data. (The columns are the QTL; the rows are the individuals.) In the sixth line, we identify the individuals that are homozygous at both QTL. (Internally, in a backcross, 1 and 2 correspond to the homozygous and heterozygous genotypes, respectively. In an intercross, 1 and 3 are the two homozygous genotypes and 2 is the heterozygous genotype.)

We might create a binary version of this phenotype by thresholding at 1. (Individuals with quantitative phenotype  $> 1$  become affected; the others are unaffected.) We can paste this into the simulated data as a second phenotype.

```
> binphe <- as.numeric(episim$pheno[,1] > 1)
> episim$pheno$affected <- binphe
```

There will now be a second phenotype named “**affected**” with 1 and 0 indicating affected and unaffected, respectively.

Finally, we might assign sexes to the individuals at random, and include a sex difference in the phenotype and even a difference in the effect of the QTL in the two sexes (a QTL  $\times$  sex interaction). We’ll create a third phenotype with these features, and place “sex” in the data as a fourth phenotype. Here, 0 and 1 correspond to females and males, respectively.

```
> sex <- sample(0:1, nind(episim), replace=TRUE)
> phe3 <- rnorm(nind(episim), 0, 1)
> phe3[wh & sex==0] <- phe3[wh & sex==0] - 1.5
> phe3[wh & sex==1] <- phe3[wh & sex==1] - 0.5
```

```
> episim$pheno$pheno3 <- phe3
> episim$pheno$sex <- sex
```

We use the R function `sample` to sample with replacement from the vector (0, 1), and `rnorm` to simulate standard normal data. The epistasis pattern for the two QTL is as before, but the effects are different in the two sexes. We reuse the `wh` object, created above, that indicated the individuals who were homozygous at both QTL.

## 2.6 Internal data structure

In this section, we describe the internal data structures used by R/qtl for cross and genetic map objects and the R syntax required to get access to the data. Other data structures (such as those produced by the `scanone` and `scantwo` functions) will be described in later chapters. This section is quite technical and will require a reasonably detailed understanding of R, and so it should probably be skipped initially. The choice of data structures required some balance between ease of programming and simplicity for the user interface. The syntax for references to certain pieces of the internal data can be quite complicated.

### 2.6.1 Experimental cross

We describe the internal data structure used by R/qtl for QTL mapping data; we will look at the data set `hyper` as an example. First, the object has a “class,” which indicates that it corresponds to data for an experimental cross, and gives the cross type. By having class “`cross`”, the functions `plot` and `summary` know to send the data to `plot.cross` and `summary.cross`.

```
> data(hyper)
> class(hyper)

[1] "bc"      "cross"
```

As you can see, the class is a two-element vector containing first a character string indicating the cross type (“`bc`” or “`f2`”) and second “`cross`” to indicate that it is an experimental cross.

Every cross object is a list with two components, one containing the genotype data and genetic maps and the other containing the phenotype data.

```
> names(hyper)

[1] "geno"  "pheno"
```

The phenotype data is simply a matrix (more strictly a data frame) with rows corresponding to individuals and columns corresponding to phenotypes. We look at the phenotypes for the first five individuals as follows.

```
> hyper$pheno[1:5,]
```

```
      bp sex
1 109.6 male
2 109.8 male
3 110.1 male
4 110.6 male
5 115.0 male
```

The first phenotype is the blood pressure of each mouse; the second phenotype indicates their sex. (In this case, all mice are male.) The phenotypes can be either numeric or factors. The sex phenotype can be coded 0/1, f/m, F/M, or female/male for female/male; in all but the first case, it must be a factor.

The genotype data is a list with components corresponding to chromosomes. Each chromosome has a name and a class. The class for a chromosome is "A" or "X", for autosomes or the X chromosome, respectively.

```
> names(hyper$geno)
```

```
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11"
[12] "12" "13" "14" "15" "16" "17" "18" "19" "X"
```

```
> sapply(hyper$geno, class)
```

```
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
"A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
16 17 18 19  X
"A" "A" "A" "A" "X"
```

Each component of `geno` is itself a list with two components, `data` (containing the marker genotype data) and `map` (containing the positions of the markers, in cM). The genotype data are coded 1/2 for homozygotes and heterozygotes in a backcross, and 1/2/3/4/5 for the genotypes AA/AB/BB/not BB/not AA in an intercross.

```
> names(hyper$geno[[3]])
```

```
[1] "data" "map"
```

```
> hyper$geno[[3]]$data[91:94,]
```

```
      D3Mit164 D3Mit6 D3Mit11 D3Mit14 D3Mit44 D3Mit19
91          2      1        1        1        1        1
92          1      1        1        1        1        1
93         NA      2       NA       NA       NA       NA
94         NA      2       NA       NA       NA       NA
```

```
> hyper$geno[[3]]$map
```

```
D3Mit164  D3Mit6  D3Mit11  D3Mit14  D3Mit44  D3Mit19
      2.2    17.5    37.2    44.8    57.9    66.7
```



On the X chromosome, all individuals are coded with genotypes 1/2. We use the phenotypes `sex` and `pgm`, if they are available, to recode these as AA/AB/BB/AY/BY before later analysis. The 1/2 codes simplify the use of the HMM algorithms (as in `calc.genoprob`, to calculate genotype probabilities), as all individuals may be treated as a backcross.

That completes the description of the raw data. However, other information may exist in a cross object, as when one runs `calc.genoprob`, `sim.geno`, or `calc.errorlod`, the output is the input cross object with the derived data attached to each component (the chromosomes) of the `geno` component.

```
> names(hyper$geno[[3]])
[1] "data" "map"
> hyper <- calc.genoprob(hyper, step=10, error.prob=0.01)
> names(hyper$geno[[3]])
[1] "data" "map" "prob"
> hyper <- sim.geno(hyper, step=10, n.draws=2, error.prob=0.01)
> names(hyper$geno[[3]])
[1] "data" "map" "prob" "draws"
> hyper <- calc.errorlod(hyper, error.prob=0.01)
> names(hyper$geno[[3]])
[1] "data" "map" "prob" "draws" "errorlod"
```

The structure of the individual components that were added is relatively self-explanatory.

Finally, when one runs `est.rf`, a matrix containing the pairwise recombination fractions and LOD scores is added to the cross object.

```
> names(hyper)
[1] "geno" "pheno"
> hyper <- est.rf(hyper)
> names(hyper)
[1] "geno" "pheno" "rf"
```

The `hyper$rf` object is a matrix. Values on the diagonal are the number of individuals that were genotyped for the corresponding marker. Values above the diagonal are LOD scores for a test of linkage; values below the diagonal are estimated recombination fractions.

```
> hyper$rf[1:4,1:4]
```

```

          D1Mit296 D1Mit123 D1Mit156 D1Mit178
D1Mit296  92.0000  11.4201   3.1422   0.6321
D1Mit123   0.1413  92.0000   9.9274   0.6321
D1Mit156   0.3043   0.1630 250.0000   2.9045
D1Mit178   0.1667   0.1667   0.2449  49.0000

```

The function `clean.cross` may be used to remove the intermediate results from a cross object (such as those created with `calc.genoprob` and `est.rf`), as follows.

```

> hyper <- clean(hyper)
> names(hyper)

[1] "geno" "pheno"

> names(hyper$geno[[3]])

[1] "data" "map"

```

## 2.6.2 Genetic map

A genetic map object, as produced by `sim.map` or as extracted from a cross object with `pull.map`, also has a somewhat complex form. We will look at the data set `map10`, a genetic map modeled after the mouse genome. Such a map object has class "map" so that `plot` and `summary` will call `plot.map` and `summary.map`, respectively.

```

> data(map10)
> class(map10)

[1] "map"

```

The map is a list whose components are the individual chromosomes. Each chromosome has class either "A" or "X" according to whether it is an autosome or the X chromosome.

```

> names(map10)

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11"
[12] "12" "13" "14" "15" "16" "17" "18" "19" "X"

> sapply(map10, class)

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
"A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
16 17 18 19 X
"A" "A" "A" "A" "X"

```

The individual chromosomes are vectors specifying the marker locations in cM, with names being the marker names.

```
> map10[[15]]  
D15M1 D15M2 D15M3 D15M4 D15M5 D15M6 D15M7 D15M8 D15M9  
 0.00 10.12 20.25 30.38 40.50 50.62 60.75 70.88 81.00  
attr(,"class")  
[1] "A"
```

## 2.7 Further reading

Broman and Heath (2007) discuss the management and manipulation of genetic data. They emphasize the need for biologists to learn to program, and the value of the Perl programming language for geneticists. While they focus on human linkage data, the general principles apply to all genetic data.

Useful Perl books include *Learning Perl* (Schwartz *et al.*, 2008) for beginners, *Programming Perl* (Wall *et al.*, 2000) as a reference, and *Perl Cookbook* (Christiansen and Torkington, 2003) for its recipes encompassing many common tasks. These books, plus a couple of others, may be purchased together on a CD for a very good price: the *Perl CD Bookshelf*, available from O'Reilly Media.

Regarding the  $\chi^2$  model for crossover interference, see Zhao *et al.* (1995). The Stahl model was described in Copenhaver *et al.* (2002).

---

## References

- Ahmadiyeh N, Churchill GA, Shimomura K, Solberg LC, Takahashi JS, Redei EE (2003) X-linked and lineage-dependent inheritance of coping responses to stress. *Mamm. Genome*, **14**:748–757.
- Baierl A, Bogdan M, Frommlet F, Futschik A (2006) On locating multiple interacting quantitative trait loci in intercross designs. *Genetics*, **173**:1693–1703.
- Baldi P, Chauvin Y, Hunkapiller T, McClure MA (1994) Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA*, **91**:1059–1063.
- Basten CJ, Weir BS, Zeng ZB (2002) QTL Cartographer: A reference manual and tutorial for QTL mapping. Program in Statistical Genetics, Bioinformatics Research Center, Department of Biostatistics, North Carolina State University.
- Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.*, **41**:164–171.
- Beavis WD (1994) The power and deceit of QTL experiments: Lessons from comparative QTL studies. In Wilkinson DB, editor, *49th Annual Corn and Sorghum Research Conference*, pages 252–268, American Seed Trade Association, Washington, DC.
- Belknap JK (1998) Effect of within-strain sample size on QTL detection and mapping using recombinant inbred mouse strains. *Behav. Genet.*, **28**:29–38.
- Bogdan M, Ghosh JK, Doerge RW (2004) Modifying the Schwartz Bayesian Information Criterion to locate multiple interacting quantitative trait loci. *Genetics*, **167**:989–999.
- Boyartchuk VL, Broman KW, Mosher RE, D’Orazio SEF, Starnbach MN, Dietrich WF (2001) Multigenic control of *Listeria monocytogenes* susceptibility in mice. *Nat. Genet.*, **27**:259–260.
- Broman KW (1999) Cleaning genotype data. *Genet. Epidemiol.*, **17**(Suppl 1):S79–83.
- Broman KW (2001) Review of statistical methods for QTL mapping in experimental crosses. *Lab Animal*, **30**(7):44–52.
- Broman KW (2003) Mapping quantitative trait loci in the case of a spike in the phenotype distribution. *Genetics*, **163**:1169–1175.

- Broman KW, Heath SC (2007) Managing and manipulating genetic data. In Barnes MR, Gray IC, editors, *Bioinformatics for Geneticists*, pages 17–31, Wiley, New York, 2nd edition.
- Broman KW, Rowe LB, Churchill GA, Paigen K (2002) Crossover interference in the mouse. *Genetics*, **160**:1123–1131.
- Broman KW, Sen S, Owens SE, Manichaikul A, Southard-Smith EM, Churchill GA (2006) The X chromosome in quantitative trait locus mapping. *Genetics*, **174**:2151–2158.
- Broman KW, Speed TP (1999) A review of methods for identifying QTLs in experimental crosses. In Seillier-Moisewitsch F, editor, *Statistics in Molecular Biology and Genetics*, volume 33 of *IMS Lecture Notes - Monograph Series*, pages 114–142, Institute of Mathematical Statistics, Hayward, CA.
- Broman KW, Speed TP (2002) A model selection approach for the identification of quantitative trait loci in experimental crosses (with discussion). *J. R. Statist. Soc. B*, **64**:641–656, 737–775.
- Broman KW, Wu H, Sen S, Churchill GA (2003) R/qtl: QTL mapping in experimental crosses. *Bioinformatics*, **19**:889–890.
- Brown T (2006) *Genomes 3*. Wiley, New York.
- Cartwright DA, Troglio M, Velasco R, Gutin A (2007) Genetic mapping in the presence of genotyping errors. *Genetics*, **176**:2521–2527.
- Chen Z (2005) The full EM algorithm for the MLEs of QTL effects and positions and their estimated variances in multiple-interval mapping. *Biometrics*, **61**:474–480.
- Christiansen T, Torkington N (2003) *Perl Cookbook*. O'Reilly Media, Sebastopol, CA, 2nd edition.
- Churchill GA (1989) Stochastic models for heterogeneous DNA sequences. *B. Math. Biol.*, **51**:79–94.
- Churchill GA, Doerge RW (1994) Empirical threshold values for quantitative trait mapping. *Genetics*, **138**:963–971.
- Copenhaver GP, Housworth EA, Stahl FW (2002) Crossover interference in arabidopsis. *Genetics*, **160**:1631–1639.
- Cox DR (1972) Regression models and life tables. *J. Roy. Stat. Soc. B*, **34**:187–220.
- Cox DR, Hinkley DV (1974) *Theoretical Statistics*. Chapman and Hall, London.
- Dalgaard P (2002) *Introductory Statistics with R*. Springer, New York.
- Darvasi A (1998) Experimental strategies for the genetic dissection of complex traits in animal models. *Nat. Genet.*, **18**:19–24.
- Darvasi A, Soller M (1992) Selective genotyping for determination of linkage between a marker locus and a quantitative trait locus. *Theor. Appl. Genet.*, **85**:353–359.
- Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Roy. Stat. Soc. B*, **39**:1–38.
- Diao G, Lin DY (2005) Semiparametric methods for mapping quantitative trait loci with censored data. *Biometrics*, **61**:789–798.
- Diao G, Lin DY, Zou F (2004) Mapping quantitative trait loci with censored observations. *Genetics*, **168**:1689–1698.
- Doerge RW, Churchill GA (1996) Permutation tests for multiple loci affecting a quantitative character. *Genetics*, **142**:285–294.
- Doerge RW, Zeng ZB, Weir BS (1997) Statistical issues in the search for genes affecting quantitative traits in experimental populations. *Stat. Sci.*, **12**:195–219.
- Draper NR, Smith H (1998) *Applied Regression Analysis*. Wiley, New York, 3rd edition.

- Dupuis J, Siegmund D (1999) Statistical methods for mapping quantitative trait loci from a dense set of markers. *Genetics*, **151**:373–386.
- Falconer DS, Mackay TFC (1996) *Introduction to Quantitative Genetics*. Prentice-Hall, Harlow, 4th edition.
- Feenstra B, Skovgaard IM, Broman KW (2006) Mapping quantitative trait loci by an extension of the Haley–Knott regression method using estimating equations. *Genetics*, **173**:2269–2282.
- Flint J, Valdar W, Shifman S, Mott R (2005) Strategies for mapping and cloning quantitative trait genes in rodents. *Nat. Rev. Genet.*, **6**:271–286.
- Fredkin DR, Rice JA (2001) Fast evaluation of the likelihood of an HMM: Ion channel currents with filtering and colored noise. *IEEE Trans. Signal Process.*, **49**:625–633.
- Gonick L, Smith W (1993) *The Cartoon Guide to Statistics*. HarperCollins, New York.
- Gonick L, Wheelis M (1991) *The Cartoon Guide to Genetics*. HarperCollins, New York.
- Grant GG, Robinson SW, Edwards RE, Clothier B, Davies RB, Judah DJ, Broman KW, Smith AC (2006) Multiple polymorphic genes determine “normal” hepatic and splenic iron status in mice. *Hepatology*, **44**:174–185.
- Hackett CA, Weller JI (1995) Genetic mapping of quantitative trait loci for traits with ordinal distributions. *Biometrics*, **51**:1252–1263.
- Haley CS, Knott SA (1992) A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. *Heredity*, **69**:315–324.
- Hastie T, Tibshirani R, Friedman J (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition.
- Henderson J, Salzberg SL, Fasman K (1997) Finding genes in human DNA with a hidden Markov model. *J. Comput. Biol.*, **4**:127–141.
- Hubbard TJ, Park J (1995) Fold recognition and ab initio structure predictions using hidden Markov models and  $\beta$ -strand pair potentials. *Proteins*, **23**:398–402.
- Ihaka R, Gentleman R (1996) R: A language for data analysis and graphics. *J. Comput. Graph. Stat.*, **5**:299–314.
- Jansen RC (1992) A general mixture model for mapping quantitative trait loci by using molecular markers. *Theor. Appl. Genet.*, **85**:252–260.
- Jansen RC (1993a) Interval mapping of multiple quantitative trait loci. *Genetics*, **135**:205–211.
- Jansen RC (1993b) Maximum likelihood in a generalized linear finite mixture model by using the EM algorithm. *Biometrics*, **49**:227–231.
- Jansen RC (2007) Quantitative trait loci in inbred lines. In Balding DJ, Bishop M, Cannings C, editors, *Handbook of Statistical Genetics*, volume 1, pages 589–622, Wiley, Chichester, 3rd edition.
- Jansen RC, Stam P (1994) High resolution of quantitative traits into multiple loci via interval mapping. *Genetics*, **136**:1447–1455.
- Jiang C, Zeng ZB (1995) Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics*, **140**:1111–1127.
- Jiang C, Zeng ZB (1997) Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. *Genetica*, **101**:47–58.
- Jin C, Lan H, Attie AD, Churchill GA, Yandell BS (2004) Selective phenotyping for increased efficiency in genetic mapping studies. *Genetics*, **168**:2285–2293.

- Johnson KR, Wright JE Jr, May B (1987) Linkage relationships reflecting ancestral tetraploidy in salmonid fish. *Genetics*, **116**:579–591.
- Kao CH, Zeng ZB (1997) General formulas for obtaining the MLEs and the asymptotic variance-covariance matrix in mapping quantitative trait loci when using the EM algorithm. *Biometrics*, **53**:653–665.
- Kao CH, Zeng ZB, Teasdale RD (1999) Multiple interval mapping for quantitative trait loci. *Genetics*, **152**:1203–1216.
- Krogh A, Brown M, Mian IS, Sjölander K, Haussler D (1994) Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, **235**:1501–1531.
- Kruglyak L, Lander ES (1995) A nonparametric approach for mapping quantitative trait loci. *Genetics*, **139**:1421–1428.
- Lander ES, Botstein D (1989) Mapping Mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics*, **121**:185–199.
- Lander ES, Green P (1987) Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci. USA*, **84**:2363–2367.
- Lander ES, Green P, Abrahamson J, Barlow A, Daly MJ, Lincoln SE, Newburg L (1987) MAPMAKER: an interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics*, **1**:174–181.
- Lange K (1999) *Numerical Analysis for Statisticians*. Springer, New York.
- Lincoln SE, Lander ES (1992) Systematic detection of errors in genetic linkage data. *Genomics*, **14**:604–610.
- Liu BH (1998) *Statistical Genomics: Linkage, Mapping, and QTL Analysis*. CRC Press, Boca Raton, FL.
- Ljungberg K, Holmgren S, Carlborg O (2002) Efficient algorithms for quantitative trait loci mapping problems. *J. Comput. Biol.*, **9**:793–804.
- Ljungberg K, Holmgren S, Carlborg O (2004) Simultaneous search for multiple QTL using the global optimization algorithm DIRECT. *Bioinformatics*, **20**:1887–1895.
- London SJ, Colditz GA, Stampfer MJ, Willett WC, Rosner B, Speizer FE (1989) Prospective-study of relative weight, height, and risk of breast-cancer. *J. Am. Med. Asso.*, **262**:2853–2858.
- Lynch M, Walsh B (1998) *Genetics and Analysis of Quantitative Traits*. Sinauer, Sunderland, MA.
- Macdonald SJ, Goldstein DB (1999) A quantitative genetic analysis of male sexual traits distinguishing the sibling species *Drosophila simulans* and *D. sechellia*. *Genetics*, **153**:1683–1699.
- Manichaikul A, Dupuis J, Sen S, Broman KW (2006) Poor performance of bootstrap confidence intervals for the location of a quantitative trait locus. *Genetics*, **174**:481–489.
- Manichaikul A, Moon JY, Sen S, Yandell BS, Broman KW (2009) A model selection approach for the identification of quantitative trait loci in experimental crosses, allowing epistasis. *Genetics*, **181**:1077–1086.
- Manichaikul A, Palmer AA, Sen S, Broman KW (2007) Significance thresholds for quantitative trait locus mapping under selective genotyping. *Genetics*, **177**:1963–1966.
- Manly KF, Cudmore RH Jr, Meer JM (2001) Map Manager QTX, cross-platform software for genetic mapping. *Mamm. Genome*, **12**:930–932.

- Martínez O, Curnow RN (1992) Estimating the locations and the sizes of the effects of quantitative trait loci using flanking markers. *Theor. Appl. Genet.*, **85**:480–488.
- McIntyre LM, Coffman CJ, Doerge RW (2001) Detection and localization of a single binary trait locus in experimental populations. *Genet. Res.*, **78**:79–92.
- McPeck MS (1996) An introduction to recombination and linkage analysis. In Speed T, Waterman MS, editors, *Genetic Mapping and DNA sequencing*, volume 81 of *IMA Volumes in Mathematics and Its Applications*, pages 1–14, Springer, New York.
- McPeck MS, Speed TP (1995) Modeling interference in genetic recombination. *Genetics*, **139**:1031–1044.
- Meng XL, Rubin DB (1993) Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, **80**:267–278.
- Miller A (2002) *Subset Selection in Regression*. Chapman & Hall/CRC, Boca Raton, FL, 2nd edition.
- Moreno CR, Elsen JM, Le Roy P, Ducrocq V (2005) Interval mapping methods for detecting QTL affecting survival and time-to-event phenotypes. *Genet. Res.*, **85**:139–149.
- Nichols KM, Broman KW, Sundin K, Young JM, Wheeler PA, Thorgaard GH (2007) Quantitative trait loci  $\times$  maternal cytoplasmic environment interaction for development rate in *Oncorhynchus mykiss*. *Genetics*, **175**:335–347.
- Nichols KM, Young WP, Danzmann RG, Robison BD, Rexroad C, Noakes M, Phillips RB, Bentzen P, Spies I, Knudsen K, Allendorf FW, Cunningham BM, Brunelli J, Zhang H, Ristow S, Drew R, Brown KH, Wheeler PA, Thorgaard GH (2003) A consolidated linkage map for rainbow trout (*Oncorhynchus mykiss*). *Anim. Genet.*, **34**:102–115.
- Orgogozo V, Broman KW, Stern DL (2006) High-resolution quantitative trait locus mapping reveals sign epistasis controlling ovariole number between two *Drosophila* species. *Genetics*, **173**:197–205.
- Owens SE, Broman KW, Wiltshire T, Elmore JB, Bradley KM, Smith JR, Southard-Smith EM (2005) Genome-wide linkage analysis identifies novel modifier loci of aganglionosis in the *Sox10<sup>Dom</sup>* model of Hirschsprung disease. *Hum. Mol. Genet.*, **14**:1549–1558.
- Purcell S, Cherny SS, Sham PC (2003) Genetic Power Calculator: design of linkage and association genetic mapping studies of complex traits. *Bioinformatics*, **19**:149–150.
- Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**:257–286.
- Reilly KM, Broman KW, Bronson RT, Tsang S, Loisel DA, Christy ES, Sun Z, Diehl J, Munroe DJ, Tuskan RG (2006) An imprinted locus epistatically influences *Nstr1* and *Nstr2* to control resistance to nerve sheath tumors in a neurofibromatosis type 1 mouse model. *Cancer Res.*, **66**:62–68.
- Rice JA (2006) *Mathematical Statistics with Data Analysis*. Duxbury Press, Belmont, CA, 3rd edition.
- Schwartz RL, Phoenix T, Foy BD (2008) *Learning Perl*. O'Reilly Media, Sebastopol, CA, 5th edition.
- Sen S, Churchill GA (2001) A statistical framework for quantitative trait mapping. *Genetics*, **159**:371–387.



- Sen S, Johannes F, Broman KW (2009) Selective genotyping and phenotyping strategies in a complex trait context. *Genetics*, **181**:1613–1626.
- Sen S, Satagopan JM, Broman KW, Churchill GA (2007) R/qtldesign: inbred line cross experimental design. *Mamm. Genome*, **18**:87–93.
- Sen S, Satagopan JM, Churchill GA (2005) Quantitative trait loci study design from an information perspective. *Genetics*, **170**:447–464.
- Sillanpää MJ, Corander J (2002) Model choice in gene mapping: what and why. *Trends Genet.*, **18**:301–307.
- Silver L (1995) *Mouse Genetics: Concepts and Applications*. Oxford University Press, Oxford.
- Solberg LC, Baum AE, Ahmadiyah N, Shimomura K, Li R, Turek FW, Churchill GA, Takahashi JS, Redei EE (2004) Sex- and lineage-specific inheritance of depression-like behavior in the rat. *Mamm. Genome*, **15**:648–662.
- Soller M, Brody T, Genizi A (1976) On the power of experimental designs for the detection of linkage between marker loci and quantitative loci in crosses between inbred lines. *Theor. Appl. Genet.*, **47**:35–39.
- Speed TP (1996) What is a genetic map function? In Speed T, Waterman MS, editors, *Genetic Mapping and DNA Sequencing*, volume 81 of *IMA Volumes in Mathematics and Its Applications*, pages 65–88, Springer, New York.
- Strickberger MW (1985) *Genetics*. Macmillan, New York, 3rd edition.
- Sugiyama F, Churchill GA, Higgins DC, Johns C, Makaritsis KP, Gavras H, Paigen B (2001) Concordance of murine quantitative trait loci for salt-induced hypertension with rat and human loci. *Genomics*, **71**:70–77.
- Symons RC, Daly MJ, Fridlyand J, Speed TP, Cook WD, Gerondakis S, Harris AW, Foote SJ (2002) Multiple genetic loci modify susceptibility to plasmacytoma-related morbidity in E $\mu$ -v-*abl* transgenic mice. *Proc. Natl. Acad. Sci. USA*, **99**:11299–11304.
- Venables WN, Ripley BD (2002) *Modern Applied Statistics with S*. Springer, New York, 4th edition.
- Visscher PM, Haley CS, Knott SA (1996a) Mapping QTLs for binary traits in back-cross and F2 populations. *Genet. Res.*, **68**:55–63.
- Visscher PM, Thompson R, Haley CS (1996b) Confidence intervals in QTL mapping by bootstrapping. *Genetics*, **143**:1013–1020.
- Wahlsten D, Metten P, Phillips TJ, Boehm SL, Burkhart-Kasch S, Dorow J, Doerksen S, Downing C, Fogarty J, Rodd-Henricks K, Hen R, McKinnon CS, Merrill CM, Nolte C, Schalomon M, Schlumbohm JP, Sibert JR, Wenger CD, Dudek BC, Crabbe JC (2003) Different data from different labs: lessons from studies of gene–environment interaction. *J. Neurobiol.*, **54**:283–311.
- Wall L, Christiansen T, Orwant J (2000) *Programming Perl*. O'Reilly Media, Sebastopol, CA, 3rd edition.
- Whittaker JC, Thompson R, Visscher PM (1996) On the mapping of QTL by regression of phenotype on marker-type. *Heredity*, **77**:23–32.
- Wu R, Ma C, Casella G (2007) *Statistical Genetics of Quantitative Traits: Linkage, Maps and QTL*. Springer, New York.
- Xu S (1998) Iteratively reweighted least squares mapping of quantitative trait loci. *Behav. Genet.*, **28**:341–355.
- Xu S, Atchley WR (1996) Mapping quantitative trait loci for complex binary diseases using line crosses. *Genetics*, **143**:1417–1424.

- Yandell BS, Mehta T, Banerjee S, Shriner D, Venkataraman R, Moon JY, Neely WW, Wu H, von Smith R, Yi N (2007) R/qtlbim: QTL with Bayesian Interval Mapping in experimental crosses. *Bioinformatics*, **23**:641–643.
- Yi N (2004) A unified Markov chain Monte Carlo framework for mapping multiple quantitative trait loci. *Genetics*, **167**:967–975.
- Yi N, Shriner D (2008) Advances in Bayesian multiple quantitative trait loci mapping in experimental crosses. *Heredity*, **100**:240–252.
- Yi N, Shriner D, Banerjee S, Mehta T, Pomp D, Yandell BS (2007) An efficient Bayesian model selection approach for interacting quantitative trait loci models with many effects. *Genetics*, **176**:1865–1877.
- Yi N, Yandell BS, Churchill GA, Allison DB, Eisen EJ, Pomp D (2005) Bayesian model selection for genome-wide epistatic QTL analysis. *Genetics*, **170**:1333–1344.
- Zeng ZB (1993) Theoretical basis for separation of multiple linked gene effects in mapping quantitative trait loci. *Proc. Natl. Acad. Sci. USA*, **90**:10972–10976.
- Zeng ZB (1994) Precision mapping of quantitative trait loci. *Genetics*, **136**:1457–1468.
- Zeng ZB, Kao CH, Basten CJ (1999) Estimating the genetic architecture of quantitative traits. *Genet. Res.*, **74**:279–289.
- Zhao H, Speed TP (1996) On genetic map functions. *Genetics*, **142**:1369–1377.
- Zhao H, Speed TP, McPeck MS (1995) Statistical analysis of crossover interference using the chi-square model. *Genetics*, **139**:1045–1056.

---

## Index

- !, **51**, 100, 288
- +.*scanone*, 195
- .*scanone*, **87**, 187, 195, 201
- .*scantwo*, 237
- ..., **26**, 27
- .*Renviron*, 358
- .*Rprofile*, 21, **358**
- <-, 26
- ?, *see* help files
  
- abline*, **87**, 187
- add.cim.covar*, 209
- addint*, 259, **266–267**, 308, 328
- additive covariate, *see* covariate, additive
- additive effect, *see* effect, additive
- addpair*, 258, **269–272**, 295, 304, 351
- addqtl*, 258, **267–269**, 294, 302, 309, 325, 342
- addtoqtl*, 259, **272**, 309, 343
- advanced intercross lines (AIL), 168
- analysis of variance (ANOVA), 76, 179, **185–186**, 261, 316
- anova*, **185**, 186, 316
- aov*, **185**, 186, 316
- apply*, **50**, 287
- args*, 24–25
- arguments, 24–26
- as.formula*, 342
- association mapping, 169
- attach*, 359
- attr*, 277
- attributes*, 277
  
- backcross*, 3, **4**, 155, 160
- barplot*, 36
- Bayes credible interval, **118**, 265, 297, 350
- Bayesian QTL mapping, 255–258
- bayesint*, **120–121**, 265, 297, 350
- bias (due to selection), 123–125
- binary trait mapping, **139–141**, 198–205, 228–231
- binom.test*, 108
- bootstrap confidence intervals, 119–122
- boxplot*, **184**, 287
  
- c*, 25, **48**
- c.scanone*, **189**, 201
- c.scanoneperm*, 223
- c.scantwoperm*, 223
- calc.errorlod*, 44, **67**
- calc.genoprob*, 44, **84**, 103, 106, 115, 137, 148, 171, 187, 201, 206, 217, 229, 234, 263, 324
- calc.penalties*, **275**, 279, 298, 304, 336
- cbind.scanoneperm*, **189**, 201
- centiMorgan (cM), **8**, 23
- ch3a* data, **47–50**, 52–53, 365
- ch3b* data, **50–51**, 365
- ch3c* data, **53–58**, 61–64, 366
- checkAlleles*, 54
- $\chi^2$  test, 50, **200**
- chisq.test*, 200
- chromosome ID, **23**, 148, 320
- chromosome substitution strains (CSS), 169

- `ci.length`, 168
- `cim`, 209
- class (of object), **34**, 35, 56
  - "`cross`", 34, 42
  - "`bc`", 42
  - "`dh`", 314
  - "`f2`", 42
  - "`map`", 45
  - "`qt1`", 258, 276
  - "`scanone`", 79, 148
  - "`scanoneboot`", 121
  - "`scanoneperm`", 106
  - "`scantwo`", 217
  - "`scantwoperm`", 222
- `clean.cross`, **45**, 300–301
- `clean.scantwo`, 224
- `col`, 70, **115**, 185
- Collaborative Cross (CC), 169
- `comment.char`, 28
- `comparegeno`, 52
- composite interval mapping (CIM), **205–206**, 209–210
- Comprehensive R Archive Network (CRAN), 18, 33, 159, **355**
- confidence interval (for QTL location), **118–122**, 168, 173, 265, 296–297, 350
- congenic strain, 3, 123–125, 169, **243**
- consomic strain, 169
- `cor`, 285
- `countX0`, 68–70
- coupling, *see* linked QTL, coupling
- covariate, 7, 113, **154**, 179, 184, 263
  - additive, **180–181**, 236, 301, 324
  - interactive, **190–192**, 237, 339–349
  - matrix, 182, **187**, 195, 207, 237, 263, 302, 324
- Cox proportional hazards model, 146–148
- cross direction, *see* direction (of cross)
- crossover interference, **12–14**, 40, 66, 243, 373
- csv format, 22–24
- csvr format, 29
- csvs format, 29–30
- csvsr format, 30
- `data`, 47
- `detach`, 359
- `detectable`, 160, 161
- diagnostics, **47–72**, 154, 284–291, 314–323
- direction (of cross), 24, **108–112**, 234
- directory (working), 25, **358–359**
- documentation, 359–360
- dominance effect, *see* effect, dominance
- doubled haploids, 313–314
- `drop.markers`, **96**, 98
- `drop.nullmarkers`, **200**, 229
- `dropfromqt1`, 259, **274**, 299, 333–335, 351
- effect
  - additive, 39, 122, **156**
  - dominance, 39, 122, **156**
  - QTL, 11, 15, 78, **122–127**, 155–156, 180, 190, 262
    - coupling, *see* linked QTL, coupling
    - in examples, 204, 224–227, 230–231, 292–294, 338–339, 351–353
    - repulsion, *see* linked QTL, repulsion
- `effectplot`, **125**, 197–198, 204, 224, 230, 292, 299
- EM algorithm, **82–83**, 139, 142, 183, 199, 217, 247, 371, 379
- Emacs Speaks Statistics (ESS), 358
- email lists, 360
- epistasis, **15–16**, 41, 78, 84, 213, 216, 243–245, 251–254, 263, 266, 277
  - in examples, 218, 221, 227, 230, 305, 328, 332, 353
- `error.var`, 160
- `est.map`, 26, 32, 56, **64**, 289, 322
- `est.rf`, 44, **53–59**, 64, 289, 317
- exporting data, 32–33
- `expression`, **87**, 91, 94, 100
- extended Haley–Knott regression, **88–90**, 93, 98–103, 198, 247, 258
- F<sub>1</sub> generation, 3
- `fill.geno`, 207
- `find.marker`, **57**, 125, 207, 225, 299, 317, 330
- `find.markerpos`, 330
- Fisher's exact test, 200
- `fisher.test`, 200
- `fitqt1`, 258, **260–263**, 293, 307, 327, 332–336, 343, 345, 347

- for, 40, **48**, 62, 99, 149, 171, 277
- formula, *see* model, formula
- forward selection, 205
- functions, 21
  
- gc, 300
- genetic map, *see* map, genetic
- genetic marker, *see* marker
- geno.crosstab, **54**, 291, 317
- geno.table, **50–51**, 316
- genotypes, 8
- gutlength data, **184–189**, 193–198, 234–235, 237–238, 366
  
- Haley–Knott regression, 83, **86–87**, 88, 97–102, 127, 137, 146, 171, 242, 246, 258, 259, 263, 323, 325
- hazard function, 146
- help files, 22, **359–360**
- help.search, 359
- heritability, 39, 77, **122**, 155, 172
- heterogeneous stock (HS), 169
- hidden Markov model (HMM), 13, 17, 81, 215, **372**
- hist, 36, **52**, 171
- hyper data, 8–9, **33**, 58–59, 67–72, 75–76, 78–79, 84–85, 87, 89, 94, 98–101, 106–108, 120–122, 125–127, 206–209, 217–227, 259–280, 366–367
  
- import data, 314
- importing data, **22–32**
- imputation, **91–94**, 98, 102, 103, 125, 127, 207, 209–210, 214, 224, 246, 247, 259, 291, 301, 371, 376–377, 379
- inbred line, 3
- individual ID, **24**, 29, 67
- info, **160**, 165
- information
  - Fisher, 158
  - genotype, 70
- install.packages, 357
- interaction
  - QTL  $\times$  covariate, *see* covariate, interactive
  - QTL  $\times$  QTL, *see* epistasis
- interaction penalty, *see* penalty
  
- interactive covariate, *see* covariate, interactive
- intercross, 3, **5**, 155, 160
- interval mapping, 80–103
- iron data, **114–118**, 127–131, 367
- is.na, 51, 288
  
- jitter, **49**, 285, 288
- jittermap, 26, **84**
  
- library, **21**, 25, 47, 358
- likelihood, 60, **76–77**, 82–83, 118, 139, 142, 215, 256, 379
- likelihood ratio, *see* LOD score
- line types, **85**, 90
- linkage group, 313
- linked QTL, 19, 78, 84, 205, 213, 224, **226**, 255, 299, 329–330, 350
  - coupling, 226
  - repulsion, **226**, 246, 249, 250, 280, 328
- listeria data, **33**, 34–36, 51, 96, 137–141, 143–146, 148–150, 367
- load, 223, **358**
- LOD profile, **264–265**, 276, 296, 308, 349, 351
- LOD score, **76–77**, 83, 86, 92–93, 137, 140, 143, 181, 191–192, 246, 250
  - genotyping errors, 66
  - linkage between markers, 53
  - marker order, 62
  - penalized, **251–254**, 274, 277, 297, *see also* penalty
  - relationship to  $F$  statistic, 77
  - relationship to heritability, 77
  - spurious, **96–97**, 109–112, 131, 135, 232, 246
  - two-QTL scan, 215–216
- LOD support interval, **118**, 172–173, 175–176, 265, 297, 350, 351
- lodint, **120–121**, 173, 265, 297, 350
- logistic regression, **198**, 228
- logit, **199**, 228
- ls, 26
- lty, **85**, 90
  
- main effects penalty, *see* penalty, main effects
- makeqt1, 258, **259**, 263, 280, 325, 342

- examples, 293, 295, 296, 303, 305, 306, 327, 333, 342
- map
  - genetic, 7–8
    - estimation, 379–380, *see also* `est.map`
    - physical, 8
- map function, **14**, 289
  - Carter–Falconer, 14
  - Haldane, 14, 173, 289
  - Kosambi, 14, 290, 320
- Map Manager QTX, **19**, 32
- map10, **37**, 38–41, 45, 170–174
- MapMaker, **18**, 30, 60, 374, 382
- marker, 7
  - density, 164–166
- marker regression, **75–78**, 83, 96
- markernames, **96**, 319, 321
- Markov chain Monte Carlo (MCMC), 249, **255–258**
- max.scanone, **79**, 267
- max.scantwo, 238
- maximum likelihood estimate (MLE), **76–77**, 82, 120, 139, 142, 199, 247, 258, 379
- memory management, 300
- mfrow, **48**, 78, 198, 204, 209, 225, 227, 231, 277, 299, 332
- minimum moment aberration (MMA)
  - method, 166
- mm format, 30–31
- mma, 166–167
- model
  - class, **244–246**, 255
  - comparison, **250–254**, 255
  - fit, 246–248
  - formula, 258, **263**, 267, 269, 270, 280, 333, 342, 343
  - search, **248–250**, 254–255
- movemarker, **57**, 319
- multiple imputation, *see* imputation
- multiple interval mapping (MIM), **247**, 258
- names, 42–45, 62, 98, 277
- nchr, **36**, 40, 62, 99
- nf1 data, **200–205**, 228–231, 368
- nind, **36**, 41, 150, 291, 324
- nmar, 36
- nmissing, **71**, 98, 189, 222, 275, 298, 304
- nonparametric interval mapping, **136–139**, 146, 198
- nphe, 36
- ntyped, **72**, 98, 107, 207, 288
- object.size, 300
- objects, 26
- optselection, 160
- optsparing, 160, **166**
- ovar data, **283–312**, 368
- pairs, 49
- par, 48
- parallel computing, **223**, 275, 297, 304
- paste, 88, 147, 287, **342**
- pch, 115
- pchisq, 202
- penalized LOD scores (pLOD), *see* LOD scores, penalized
- penalty
  - heavy interaction, 252
  - light interaction, 252–253
  - main effects, 251
- permutation test, **105–106**, 127, 135, 136, 251, 252, 306
  - in examples, 106–108, 116–117, 130–131, 138–141, 144–145, 149–150, 189, 193–197, 201–203, 207, 222–223, 229, 275, 297–298, 303–304, 324, 328, 339–340
  - number of replicates, **105–106**, 114
  - stratified, **105**, 114, 118
  - two-QTL scan, 216
  - with covariates, **182**, 192–193
  - X chromosome, 113–114
- pgm phenotype, **24**, 44, 113, 114
- pheno.col, *see* scanone, pheno.col
- phenotypes, 7, 47, **153–154**
- physical map, *see* map, physical
- pleiotropy, 127
- plot, **34**, 42, 56, 79
- plot.cross, **35**, 285, 315
- plot.geno, 24, **67**, 291
- plot.info, 70–71
- plot.map, 36, 45, 56, **64**, 289, 322
- plot.missing, 36
- plot.pheno, **36**, 48

- plot.png, 78, **126**, 224, 299, 331
- plot.qtl, 260
- plot.rf, **55**, 64, 289, 317
- plot.scanone, **79**, 85, 90, 115, 129, 148, 267
- plot.scanoneperm, 106
- plot.scantwo, **217**, 272
- plotLodProfile, 258, **264**, 276, 296, 308, 349
- plotModel, 277
- power (to detect QTL), 113, 123, 127, 161, **173**, 184, 205, 206, 216, 255
- powercalc, **159**, 173
- print, **171**, 272, 275, 298, 304, 336, 346
- probit, 199
- pull.geno, **55**, 167, 207
- pull.map, 37, **54**, 319–321, 350
- pull.pheno, **140**, 144, 149, 186, 200, 229, 237, 286, 288, 301, 324
  
- qchisq, 341
- QTL Archive, 34
- QTL Cartographer, **19**, 31
- QTL effect, *see* effect, QTL
- QTL formula, *see* model, formula
- QTL object, **258**, 259, 263, 272–274, 276, 280, 293, 297, 325, 342, 351
- qtl package, *see* R/qtl
- qtlbook package, *see* R/qtlbook
- qtlcart format, 31–32
- qtlDesign package, *see* R/qtlDesign
- qtx format, 32
- quantile, 171
  
- R, 17–18
- R/qtl, 17–18
  - web page, 355
- R/qtlbook, 33
- R/qtlDesign, 159
- raw file, 30
- rbind, 39
- read.cross, **22–32**, 314
- read.table, 26, **28**
- recombinant congenic strains (RCS), 169
- recombinant inbred lines (RILs), 4, **6**, 155, 160, 163–164
- recombination fraction, 53
  - estimation, *see* est.rf
- refineqtl, 258, **263–264**
  - examples, 296, 299, 306, 309, 327, 333, 345, 347, 349
- reorderqtl, 259, **274**, 309
- replace.map, 26, 32, **65**, 323
- replaceqtl, 259, **273**
- repulsion, *see* linked QTL, repulsion
- ripple, **60–64**, 320
- rm, 300
- rnorm, 42
- rug, 52, **171**, 174
  
- sample, **42**, 50, 167
- sample size, 155, **162–164**
- samplesize, 160, **162–164**
- sapply, 43, 45, **62–63**
- save, 223, **358**
- save.image, 359
- scanone, 78, **84**, 87, 89, 94, 115, 137, 140, 143, 167, 171, 173, 292
  - covariates, 187, 193, 201, 207, 324, 339
  - multiple phenotypes, 127
  - permutation test, **106**, 116, 130, 138, 145, 189, 194, 201, 207, 324, 340
  - pheno.col, **127–128**, 140, 144
- scanone.cph, 147–150
- scanoneboot, 121
- scanqtl, **258**, 270
- scantwo, **217**, 229, 234, 329
  - covariates, **237**
  - permutation test, **222–223**, 229, 275, 298, 304, 328
- search, 359
- segregation distortion, 12, **50–51**, 317
- selection bias, 123–125
- selective genotyping, 10, 58, 87, 97–101, 105, **165–166**, 184, 187, 207, 222, 247, 259, 275, 283, 288, 291, 298, 312
- selective phenotyping, **166–167**, 283, 296, 312
- set.seed, **193**, 201, 223, 324, 340
- significance threshold, **104–106**, 130, 146, 161, 163, 170, 189, 192, 193, 298
  - two-QTL scan, **216–217**, 222
  - X chromosome, **113–114**, 235
- sim.cross, 36, **38–41**, 167, 171, 172

- `sim.geno`, 44, **94**, 103, 125, 197, 204, 224, 230, 259, 291, 300
- `sim.map`, **37–38**, 167
- simulation
  - cross, *see* `sim.cross`
  - genetic map, *see* `sim.map`
- source, 148
- spurious LOD score, *see* LOD score, spurious
- `stepwiseqtl`, 259, **274–280**, 298, 310, 336
- stratified permutation test, *see* permutation test, stratified
- `subset.cross`, **100**, 150, 167, 184, 195, 202, 225, 229, 288, 301
- summary, **34**, 42, 79
- `summary.cross`, **34–35**, 84
- `summary.fitqtl`, **260–262**, 327
- `summary.map`, **38**, 45, 171
- `summary.ripple`, 61
- `summary.scanone`, **79**, 108, 115, 117, 148
  - format, **129**, 145, 189, 341
  - multiple phenotypes, 128–129
- `summary.scanoneperm`, 106
- `summary.scantwo`, 217, **220–222**, 223, 272, 295
- `summary.scantwoperm`, **222**, 229, 275
- survival package, 146
- `switch.order`, **62**, 320
- `system.time`, 102–103
- t* test, **76**, 92, 287
- `t.test`, 288
- `tapply`, **200**
- thresh, 160, **161**, 163, 171
- threshold, *see* significance threshold
- `top.errorlod`, 67
- `totmar`, **36**, 317
- transformation, 16, **135**, 228
- transgressive QTL, 262
- tree (regression), 245
- trout data, **313–354**, 369
- two-part model, **141–146**, 199
- `update.packages`, 357
- website
  - CRAN, 355
  - for book, **IX**, 33, 174
  - for R, 17
  - for R/qtl, 355
  - QTL archive, 34
- working directory, *see* directory (working)
- workspace, 26, 32, 223, 300, **358–359**
- `write.cross`, 33
- X chromosome, **108–118**, 232–235