Statistics for the Australian Grains Industry Technical Report Series

A vignette for the wgaim R package

Julian Taylor
Biometrician, SAGI
PMB 1 Waite Campus
University of Adelaide
Glen Osmond, 5064
email: julian.taylor@adelaide.edu.au

January 30, 2015

Contents

1 Introduction

1.1 Background & Computational history

Whole genome analysis is receiving wide attention in the statistical genetics community. In the context of plant breeding experiments the focus is on quantitative trait loci (QTL) which attempt to explain the link between a trait of interest and the underlying genetics of the plant. Many approaches of QTL analysis are available such as marker regression methods (Hayley & Knott, 1992; Martinez & Curnow, 1992) and interval mapping (Zeng, 1994; Whittaker et al., 1996). These methods are common place in QTL software and are available for use in R packages such as Karl Bromans **qtl** package (Broman & Wu, 2014). This particular suite of software is also complemented with a book (Broman & Sen, 2009) which has been favourably reviewed (Zhou, 2010).

There has also been some focus on the use of numerical integration techniques for the analysis of QTL. Xu (2003) and Zhang et al. (2008) suggest the use of Bayesian variable shrinkage and utilise Markov chain Monte Carlo (MCMC) to perform the analysis. An MCMC approach is also adopted in the R package **qtlbim** (Yandell et al., 2005). The package builds on the **qtl** package and the Bayesian paradigm allows an extensible list of trait types to be analysed. The package also makes use of the new model selection technique, the Deviance Information Criterion (Shriner & Yi, 2009), to aid in identifying the correct QTL model. Similarly, a non-MCMC approach is adopted in the **BayesQTLBIC** package (Ball, 2010) where the QTL analysis involves the use of the Bayesian Information Criterion (Schwarz, 1978) as a QTL model selection tool.

Unfortunately many of the aforementioned methods and their software lack the ability to account for complex extraneous variation usually associated with plant or animal based QTL studies. Limited covariate additions are possible in R package **qtlbim** and through the inventive on-line **GridQTL** software which uses the ideas of Seaton et al. (2002). Kang et al. (2008) uses linear mixed models in the R package **EMMA** but it does not allow for extraneous random effects and possible complex variance structures that may be needed to capture environmental processes, such as spatial layouts, existing in the experiment.

1.2 WGAIM and software package

In this vignette we discuss the whole genome average interval mapping (WGAIM) approach of Verbyla et al. (2007) and its related software, the R package **wgaim**. The package can be downloaded from the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=wgaim. This approach allows the simultaneous modelling of genetic and non-genetic variation through extensions of the linear mixed model. The extended model allows complex extraneous variation to be captured as well as simultaneously incorporating a whole genome analysis to detection and selection of QTL using a linkage map. The underlying linear mixed modelling analysis is performed computationally using the R package **ASReml-R**. The simulation results in Verbyla et al. (2007) show that WGAIM is a powerful tool for QTL detection and outperforms more rudimentary methods such as composite interval mapping. As it incorporates the whole genome into the analysis it eliminates the necessity for piecemeal model fitting along the genome which in turn avoids the use of model selection criteria or thresholding to control the number of false positive QTL. In **wgaim** the false positives are controlled naturally by assuming a background level of QTL variation through a single variance component associated with a contiguous set of QTL across the whole genome. This parameter can then be tested to determine the presence of QTL somewhere on the genome. As a result, a less cumbersome approach to detecting and selecting QTL is ensured.

1.3 Software Prerequisites

The WGAIM method uses an extension of interval mapping to perform its analysis. For convenience and flexibility, the **wgaim** package provides the ability to convert genetic data objects created in the **qtl** package to objects for use in **wgaim**. The converted objects retain a similar structure to ones created in **qtl** and therefore can still be used with functions within the package. Users of **wgaim** need to be aware that it is a software package intended for the analysis and summary of QTL and currently only contains minimal tools for exploratory linkage map manipulation. Much of the exploratory work can be handled with functions supplied in the **qtl** package and users should consult its documentation if required. In addition, the interval mapping approach of Verbyla et al. (2007) and its implementation in **wgaim** is also restricted to populations with only two distinct genotypes. Some of these populations include, doubled haploid (DH), back-crosses and recombinant inbred lines (RIL). To ensure this rule is adhered to, error trapping has been placed in the appropriate functions.

Throughout the WGAIM procedure the underlying linear mixed model analysis uses the highly flexible R software package **ASReml-R**, built as a front end wrapper for the stand alone version, **ASReml** (Gilmour et al., 2009). This software allows the user the ability to flexibly model spatial or environmental variation as well as possible variation that may arise from additional components associated with the experimental design. It uses

an average information algorithm developed in Gilmour et al. (1995) that allows efficient computing of residual maximum likelihood (REML) (Patterson & Thompson, 1971) estimates for the variance parameters. The use of REML estimation in the linear mixed model context becomes increasingly necessary in situations where the data is unbalanced. Much of its sophistication has been influenced from its common use in the analysis of crop variety trials (Smith et al., 2001, 2005, 2006) where complex additional components such as spatial correlation structures or multiplicative factor analytic models need to be incorporated into the mixed model. If available, the software also allows complex pedigree information to be included (Oakey et al., 2006). Many of these additional flexibilities in ASReml have also established it as a valuable software tool in the livestock industries. In more recent years it has been used as a core engine for more complex genetic analyses as in Gilmour (2007), Verbyla et al. (2007) and Huang & George (2009). If you are affiliated with an academic institution, the stand alone software and the R package **ASReml-R** Discovery is now freely available through `http://www.vsni.co.uk`.

2 WGAIM: Theory

2.1 WGAIM Method

The WGAIM approach is a forward selection method that uses a whole genome approach to genetic analysis at each iteration. Following Verbyla et al. (2007), initially a working model is developed that assumes a QTL in every interval. Thus for a given set of trait observations $\boldsymbol{y} = (y_1, \ldots, y_n)$ consider the model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_e\boldsymbol{u}_e + \boldsymbol{Z}_g\boldsymbol{g} + \boldsymbol{e}, \tag{2.1}$$

where $\boldsymbol{\tau}$ is a t length vector of fixed effects with an associated $n \times t$ explanatory design matrix $\boldsymbol{X}$ and $\boldsymbol{u}_e$ is a $b \times 1$ length vector of random effects with an associated $n \times b$ design matrix $\boldsymbol{Z}_e$. Typically, the distribution of $\boldsymbol{u}_e \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{G}(\boldsymbol{\varphi}))$ and is assumed mutually independent to the residual vector $\boldsymbol{e} \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{R}(\boldsymbol{\phi}))$ with $\boldsymbol{\varphi}$ and $\boldsymbol{\phi}$ being vectors of variance ratios.

The vector $\boldsymbol{g}$ in (2.1) represents a r length vector of genotypic random effects with its associated design matrix $\boldsymbol{Z}_g$. Let m be the total number of markers, c be the number of chromosomes, m_k the number of markers on chromosome k, $(k = 1, \ldots, c)$, and $q_{i,k:j}$ represent the parental allele type for line i in interval j on chromosome k. In WGAIM, $q_{i,k:j} = \pm 1$, reflecting two possible genotypes AA, BB for DH and RIL and AB, BB for back-cross populations. The ith genetic component of this model is then given by

$$g_i = \sum_{k=1}^{c} \sum_{j=1}^{m_k - 1} q_{i,k:j} a_{k:j} + p_i, \tag{2.2}$$

where $a_{k:j}$ is QTL effect size assumed to have distribution $a_{k:j} \sim N(0, \sigma^2\gamma_a)$ and $p_i \sim N(0, \sigma^2\gamma_p)$ represents a polygenic or residual genetic effect not captured by the QTL effects.

As in interval mapping the vector of QTL allele types are replaced by the expectation of the QTL genotype given the flanking markers. Let $\boldsymbol{m}_{k:j}$ be the jth marker on the kth

chromosome then the vector of genotypic effects is

$$\begin{aligned} \boldsymbol{g} &= \sum_{k=1}^{c} \sum_{j=1}^{m_k-1} (\boldsymbol{m}_{k:j}\lambda_{k:j,j} + \boldsymbol{m}_{k:j,j+1}\lambda_{k:j+1,j})a_{k:j} + \boldsymbol{p} \\ &= \boldsymbol{M}\boldsymbol{\Lambda}\boldsymbol{a} + \boldsymbol{p}, \end{aligned} \tag{2.3}$$

where $\lambda_{k:j,j}$ and $\lambda_{k:j+1,j}$ are complicated expressions based on recombination fractions between the marker and the QTL in the jth interval (see equation (5) and (6) on page 100 of Verbyla et al. (2007)). These parameters require estimation. Verbyla et al. (2007) suggest applying a parameter reduction technique to produces a vector of genotypic effects of the form

$$\begin{aligned} \boldsymbol{g} &= \sum_{k=1}^{c} \sum_{j=1}^{m_k-1} (\boldsymbol{m}_{k:j} + \boldsymbol{m}_{k:j,j+1})\lambda_{k:j}a_{k:j} + \boldsymbol{p} \\ &= \boldsymbol{M}\boldsymbol{\Lambda}_E\boldsymbol{a} + \boldsymbol{p}, \end{aligned} \tag{2.4}$$

where $\lambda_{k:j} = \theta_{k:j,j+1}/2d_{k:j,j+1}(1-\theta_{k:j,j+1})$ and $\theta_{k:j,j+1}, d_{k:j,j+1}$ are the the **known** recombination fraction and Haldane's genetic distance between marker j and $j+1$ respectively on the kth chromosome. Let $\boldsymbol{M}_E = \boldsymbol{M}\boldsymbol{\Lambda}_E$ then $\boldsymbol{M}_E$ is an $r \times (m-c)$ fully specified known matrix of pseudo-markers spanning the whole genome. A more detailed overview of this decomposition and its derivation can be found in Verbyla et al. (2007). The full working statistical model for analysis is then

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_e\boldsymbol{u}_e + \boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a} + \boldsymbol{Z}_g\boldsymbol{p} + \boldsymbol{e}. \tag{2.5}$$

After the fitting of (2.5) the simple hypothesis $H_0 : \gamma_a = 0$ is tested based on the statistic $-2\log\boldsymbol{\Psi} = -2(\log L - \log L_0)$ where L and L_0 is the residual likelihood of the working model (2.5) with and without the random regression QTL effects, $\boldsymbol{Z}_a\boldsymbol{a}$. Stram & Lee (1994) suggest that under H_0, $-2\log\boldsymbol{\Psi}$ is distributed as the mixture $\frac{1}{2}(\chi^2_0 + \chi^2_1)$ due to the necessity of testing whether the variance ratio is on the boundary on the parameter space.

If γ_a is found to be significant a putative QTL is determined using an outlier detection method based on the alternative outlier model (AOM) for linear mixed models from Gogel (1997) and formalised in Gogel et al. (2001). Verbyla et al. (2007) uses the AOM to develop a score statistic for each of the chromosomes. For example, for the kth chromosome let $\boldsymbol{a}_{k0} = \boldsymbol{a}_k + \boldsymbol{\delta}_k$ where $\boldsymbol{\delta}_k$ is a vector of random effects such that $\boldsymbol{\delta}_k \sim N(0, \sigma^2\gamma_{a,k}\boldsymbol{I}_{m_k-1})$. The full outlier model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_e\boldsymbol{u}_e + \boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a} + \boldsymbol{Z}_{g,k}\boldsymbol{M}_{E,k}\boldsymbol{\delta}_k + \boldsymbol{Z}_g\boldsymbol{p} + \boldsymbol{e}, \tag{2.6}$$

where $\boldsymbol{Z}_{g,k}$ is the matrix $\boldsymbol{Z}_g$ appropriately subsetted to chromosome k. The REML score is then derived for $\gamma_{a,k}$ and evaluated at $\gamma_{a,k} = 0$, namely

$$U_k(0) = -\frac{1}{2}\left(\mathrm{tr}(\boldsymbol{C}_{k,k}) - \frac{1}{\sigma^2\gamma_a^2}\tilde{\boldsymbol{a}}_k^T\tilde{\boldsymbol{a}}_k\right), \tag{2.7}$$

where $\boldsymbol{C}_{k,k} = \boldsymbol{Z}_{g,k}\boldsymbol{M}_E\boldsymbol{P}\boldsymbol{M}_E\boldsymbol{Z}_{g,k}$ with $\boldsymbol{P} = \boldsymbol{H}^{-1} - \boldsymbol{H}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{H}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{H}^{-1}$, $\boldsymbol{H} = \sigma^2(\boldsymbol{R}+\boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^T+\gamma_a\boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{M}_E^T\boldsymbol{Z}_a^T+\gamma_p\boldsymbol{Z}_p\boldsymbol{Z}_p^T)$ and best linear unbiased predictors (BLUPS) $\tilde{\boldsymbol{a}}_k = \gamma_a\boldsymbol{M}_E^T\boldsymbol{Z}_{g,k}^T\boldsymbol{P}\boldsymbol{y}$. This score has mean zero and this will occur exactly when the terms in the parentheses of (2.7) are equal. Scores that depart from zero suggest a departure from $\gamma_{a,k} = 0$. A simple statistic that reflects this departure can be based on the "outlier" statistic

$$t_k^2 = \frac{\tilde{\boldsymbol{a}}_k^T\tilde{\boldsymbol{a}}_k}{\sigma^2\gamma_a^2\text{tr}(\boldsymbol{C}_{k,k})} = \frac{\sum_{j=1}^{m_k-1}\tilde{a}_{k:j}^2}{\sum_{j=1}^{m_k-1}\text{var}(\tilde{a}_{k:j})}. \tag{2.8}$$

This statistic can therefore be calculated from the BLUPS of the QTL sizes and their prediction error variances arising from the working model. In most cases mixed model software, including **ASReml-R** used in **wgaim**, provide the ability to extract these components for this use.

In a similar manner to the above once the chromosome with the largest outlier statistic is identified, the individual intervals within that chromosome are checked. For example if the largest t_k^2 is from the kth chromosome, a similar derivation can be followed for the outlier statistic of the jth interval, namely

$$t_{k:j}^2 = \frac{\tilde{a}_{k:j}^2}{\text{var}(\tilde{a}_{k:j})}. \tag{2.9}$$

A putative QTL is then determined by choosing the largest $t_{k:j}^2$ within that chromosome. It must be stated at this point that although (2.6) is formulated to derive the theory for QTL outlier detection there is no requirement to fit this model as the chromosome and interval outlier statistics only contain components obtainable from a fit of the working model proposed in (2.5). Thus there is only a minimal computational cost to determine an appropriate QTL interval using this method.

Once a QTL interval is selected it is moved into the fixed effects of the working model (2.5) and the process is repeated until γ_a is not significant. After the selection process is complete the selected QTL intervals appear as fixed effects and the final model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_g\boldsymbol{M}_{E,s}\boldsymbol{a}_s + \boldsymbol{Z}_e\boldsymbol{u}_e + \boldsymbol{Z}_g\boldsymbol{M}_{E,-s}\boldsymbol{a}_{-s} + \boldsymbol{Z}_g\boldsymbol{p} + \boldsymbol{e}, \tag{2.10}$$

where $\boldsymbol{M}_{E,s}$ contain the the appropriate columns of $\boldsymbol{M}_E$ for the selected QTL with $\boldsymbol{a}_s$ as fixed effects and $\boldsymbol{M}_{E,-s}$ contain the columns of genetic information for the unselected QTL with $\boldsymbol{a}_{-s}$ as a set of random effects. The preservation of the unselected QTL component in the model ensures the selected QTL are tested within the appropriate stratum of the hierarchical model. This complete approach is known as the WGAIM algorithm.

2.2 Marker vs Interval

The WGAIM method derived in the previous section uses a whole genome extension of interval mapping. The matrix $\boldsymbol{\Lambda}$ in (2.3) can be viewed as a mapping matrix that

appropriately maps the marker scores to midpoint pseudo-interval scores. In fact, the genetic model proposed in (2.3) can be written as an approximate marker QTL regression model

$$\boldsymbol{g} = \boldsymbol{M}\boldsymbol{a}_M + \boldsymbol{p} \tag{2.11}$$

where the marker QTL sizes are $\boldsymbol{a}_M = \boldsymbol{\Lambda}\boldsymbol{a}$. This suggests the marker QTL sizes have an assumed distribution of the form $\boldsymbol{a}_M \sim N(\boldsymbol{0}, \sigma^2\gamma_a\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T)$ and are correlated. Therefore an analysis assuming the genetic QTL model (2.11) with independent marker QTL effects will be less efficient than the interval mapping equivalent (2.3). Whole genome marker or interval analysis is possible with **wgaim**.

2.3 Recent advances: WGAIM v1.0+

WGAIM is always being developed to improve its efficiency and stability as well as advance its capabilities. Recent research by Verbyla et al. (2012) has shown the WGAIM method can be improved in several ways. These are outlined below.

2.3.1 The outlier statistics

There is a short relevant point in Verbyla et al. (2012) concerning the use of the outlier statistics in the WGAIM algorithm. After much scrutiny it was found that the use of the chromosome statistic was flawed for small linkage groups. Consider the scenario of two markers on a linkage group k. After converting the marker information to a single interval the chromosome and interval outlier statistics have the property $t^2_k = t^2_{k:1}$. Thus, the chromosome statistic, in this instance, is based on the information contained in one interval. This interval statistic, in some circumstances, may bias the choice toward chromosome k and the selection of its only interval. Through simulation Verbyla et al. (2012) shows that a better choice would be to only use the interval outlier statistic to guide the selection process.

2.3.2 High dimensional analysis

Verbyla et al. (2012) show that provided there is some replication of genotypic individuals existing in the data, high dimensional genetic marker components can be included in the formulation of the working model (2.5). In fact, if the number of markers or intervals exceeds the number of genetic individuals then a transformation is always warranted. This ensures the maximum number of columns of marker or interval related information in the working WGAIM model is equal to the number of genetic individuals. As expected, this reduces computation times considerably for high dimensional problems. Further details can be found in Verbyla et al. (2012).

2.3.3 A random effects formulation

It is well known that there is (selection) bias involved in moving the selected QTL to the fixed effects (Beavis, 1994, 1998). Xu (2003) provides a theoretical justification while Melchinger et al. (1998) also conclude that sizes are inflated. There is a parallel in general plant breeding analysis where genetic effects are assumed to be random rather than fixed. This reduces the bias through shrinkage and provides a more realistic estimate of the size of a genetic effect. Reducing the bias in QTL analysis would be desirable.

In a random effects formulation we assume that the ith selected QTL appearing in the final model (2.10) has an assumed distribution $a_i \sim N(0, \sigma^2_{a_i})$. That is, the size of the QTL effects are assumed to be random and have a different variance to the unselected effects. This makes sense as a putative QTL effects exhibit variation from zero because they are QTL. Thus individual QTL have their own distribution and non-QTL come from another distribution. The two distributions differ in their variances and not their means.

2.4 Summary assessment of QTL

2.4.1 Fixed effects formulation

A summary of the additive QTL fixed effect can be obtained by considering an appropriate hypothesis test. Let $\hat{a}_{kj}$ be the fixed effect estimate of the QTL a_{kj} with variance $\text{var}(\hat{a}_{kj}) = \sigma^2_{PEV,kj}$. The test then considers the null hypothesis $H_0 : a_{kj} = 0$ against the alternative hypothesis $H_a : a_{kj} \neq 0$. A z-statistic for this QTL is then calculated using

$$z_{kj} = \frac{\hat{a}_{kj}}{\sigma_{PEV,kj}}$$

and therefore a p-value for the hypothesis test is

$$1 - \Pr\big(-z_{kj} < Z < z_{kj}\big)$$

LOD scores are generally not necessary for WGAIM but can be calculated using

$$LOD_{kj} = \frac{1}{2}\log_{10}\exp(z^2_{kj})$$

2.4.2 Random effects formulation

In this formulation, the size of the QTL effect is a best linear unbiased prediction (BLUP). It is no longer appropriate to test the hypothesis that the effect is zero in order to assess its significance. Tests of hypotheses pertain to unknown parameters, and random effects involve distributions of effects.

To provide a measure of the strength of a QTL, the conditional distribution of the true (random) QTL effect a_{kj} say, given the data is used. That is under the normality assumptions for a linear mixed model,

$$a_{kj}|\boldsymbol{y}_2 \sim N(\tilde{a}_{kj}, \sigma^2_{PEV,kj})$$

where $\boldsymbol{y}_2$ is the component of the data free of fixed effects (Verbyla, 1990). The mean of this conditional distribution is the BLUP of a_{kj}, that is the estimated size of the QTL $\tilde{a}_{kj}$, and in this instance, the variance $\sigma^2_{PEV,kj}$ is the prediction error variance (PEV) of a_{kj}. Thus the proper assessment of the impact of the QTL involves determining how far the distribution is from zero. This can be quantified by calculating a probability somewhat like a p-value, but for which values close to 0 indicate the QTL is strong. Consider the statistic

$$X^2_{kj} = \left(\frac{a_{kj} - \tilde{a}_{kj}}{\sigma_{PEV,kj}}\right)^2$$

which has a chi-squared distribution with one degree of freedom. Zero on the original scale is $c^2_{kj} = \tilde{a}^2_{kj}/\sigma^2_{PEV,kj}$ on the chi-squared scale and therefore

$$\Pr(X^2_{kj} > c^2_{kj})$$

provides a measure of strength of the putative QTL by how far $\tilde{a}_{kj}$ is away from zero relative to $\sigma_{PEV,kj}$. In a similar manner to the fixed effects formulation a LOD score can be calculated using $LOD_{kj} = \log_{10}\exp(c^2_{kj})/2$.

2.4.3 Genetic variance contribution of QTL

It is often of interest to calculate the genetic variance contribution of the selected putative QTL. This requires the total genetic variance of the genetic effects expression

$$\boldsymbol{g} = \boldsymbol{M}_{E,s}\boldsymbol{a}_s + \boldsymbol{M}_{E,-s}\boldsymbol{a}_{-s} + \boldsymbol{p}$$

Following Verbyla et al. (2012), to facilitate an expression for the variance the first term on the RHS is replaced by (2.2). For a single line i the variance then becomes

$$\text{var}(g_i) = \sum_{l=1}^{s} a_l^2 + \sum\sum_{l=l'}(1 - 2\theta_{ll'})a_l a_{l'} + \sigma^2_a \boldsymbol{m}^T_{Ei,-s}\boldsymbol{m}_{Ei,-s} + \sigma^2_p$$

where $\boldsymbol{m}_{Ei,-s}$ is the ith row of $\boldsymbol{M}_{E,-s}$. Using an average line effect, $m_{E,-ss} = \boldsymbol{m}^T_{Ei,-s}\boldsymbol{m}_{Ei,-s}/r$ and ignoring covariances between QTL the total variance across all lines is

$$\text{var}(g^*) = \sum_{l=1}^{s} a_l^2 + \sigma^2_a m_{E,-ss} + \sigma^2_p$$

The percentage contributions of the lth QTL to the genetic variance is then

$$PV_l = 100\frac{a_l^2}{\text{var}(g^*)}$$

Numerical calculations of the contributions are based on estimates of the parameters obtained from the final QTL model.

3 The R package wgaim: A casual walk through

A typical QTL analysis with **wgaim** can be viewed as series of steps with the appropriate functions

Step 1. Fit a base `asreml()` model

Fit a base `asreml()` model as in (2.5) but without the added marker/interval genetic information term $\boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a}$ using

```
baseModel <- asreml(..., data = phenoData)
```

(see the **ASReml-R** package for arguments `...`). The `asreml()` call allows very complex structures for the variance matrices $\boldsymbol{G}(\boldsymbol{\varphi})$ and $\boldsymbol{R}(\boldsymbol{\phi})$ through its `random` and `rcov` arguments. This makes it an ideal modelling tool for capturing non-genetic variation, such as design components and/or extraneous environmental variation.

For a comprehensive overview of the **ASReml-R** package, including thorough examples of its flexibility, users should, in the first instance, consult the documentation that is included with the package. **Note: On any operating system that has ASReml-R installed, the documentation can be found using the simple command `asreml.man()` in R.**

Step 2. Read in genetic data using `read.cross()`

Read in genetic data using

```
crossObj <- read.cross(...)
```

(see the **qtl** package for arguments `...`). This function allows the reading in of genetic information in a number of formats including files generated from commonly used genetic

software programs such as Mapmaker and QTL Cartographer. At the current printing of this document `read.cross()` accepts data in the following formats (from the help for `read.cross()`),

- comma-delimited ("csv")
- rotated comma-delimited ("csvr")
- comma-delimited with separate files for genotype and phenotype data ("csvs")
- rotated comma-delimited with separate files for genotype and phenotype data ("csvsr")
- Mapmaker ("mm")
- Map Manager QTX ("qtx")
- Gary Churchill's format ("gary")
- Karl Broman's format ("karl").

For the exact requirements of all available file types and their nomenclature users should consult the **qtl** documentation. The `read.cross()` function can also process more advanced genetic crosses. However, in **wgaim** the QTL analysis is restricted to populations with two genotypic states. Thus users should be aware that the class of the cross object needs to inherit one of `"bc","dh","riself"`. This is checked when converting the object in step 3.

The function `read.cross()` will also estimate map distances if they are not given in the genetic file(s) before importation. It uses the Lander & Green (1987) hidden Markov model for its estimation. This is an EM algorithm and therefore suffers from linear convergence. On some occasions the algorithm may slowly converge or not converge at all. In these instances users may need to investigate possible problems with their linkage map before attempting to import.

Step 3. Convert genetic `"cross"` object to an `"interval"` object

This can be done using the `wgaim` function

```
intervalObj <- cross2int(crossObj, missgeno = "MartinezCurnow",
               rem.mark = TRUE, id = "id", subset = NULL)
```

The function contains a number of arguments that provide some linkage map manipulation before calculation of the interval information for each chromosome. They are detailed as follows,

1 **Sub-setting**: The map can be subsetted by giving the `subset` argument a character string vector of chromosome names.

Table 3.1: Consensus marker outcomes for 3 markers in a doubled haploid population. The consensus marker uses the name of the first marker in the set prefixed with a "(C)".

Marker1	Marker2	Marker3	Marker1(C)
AA	AA	AA	AA
BB	BB	BB	BB
AA	NA	AA	AA
BB	NA	BB	BB
AA	NA	NA	AA
BB	NA	NA	BB
NA	NA	NA	NA
AA	BB	AA	NA

2 **Co-locating markers**: If `rem.mark = TRUE` then consensus markers are formed for co-locating marker sets across the genome. This is achieved by combining markers scores in the same marker set using the rules of Table 3.1. These rules have an obvious extension to larger co-located marker sets. The final consensus marker uses the name of the first marker with a "(C)" prefix to ensure the interpretation remains simple post analysis. The markers involved in the formation of each consensus markers, and their connections with one another, are returned as a named element of `intervalObj` called `"cor.markers"`.

3 **Missing values**: If `missgeno = "MartinezCurnow"`, missing values within a chromosome are imputed using the rules of Martinez & Curnow (1992). If `missgeno = "Broman"` the they are calculated using the default values of `argmax.geno()` in the **qtl** package

Note: This step is crucial in the process of QTL analysis using wgaim. The imputation of the missing markers ensures the genetic data being passed into `wgaim.asreml()` in the next step is a complete (i.e. no missing values) across all linkage groups.

After the linkage map manipulation, for each chromosome, the imputed marker data matrix is returned as an element of `intervalObj`. Along with this, several interval calculations are returned such as distances between markers, recombination fractions and most importantly, the interval data matrix, $\boldsymbol{M}_E$ defined shortly after (2.4).

The `id` argument is required to determine the unique rows of the genotypic data and is passed to the imputed marker data and the interval data matrix. The final genetic data object returned also retains the original class of the object for backward compatibility with other functions in the **qtl** package as well as inherits the class `"interval"` for functionality within the **wgaim** package.

Step 4. Perform QTL analysis with `wgaim()`

```
QTLmodel <- wgaim(baseModel, phenoData, intervalObj, merge.by = NULL,
              gen.type = "interval", method = "fixed", selection = "interval",
              breakout = -1, TypeI = 0.05, attempts = 5, trace = TRUE,
              verboseLev = 0, ...)
```

The `baseModel` argument must be an `asreml.object` and therefore have `"asreml"` as its class attribute. Thus a call to `wgaim()` is actually a call to `wgaim.asreml()`. This stipulation ensures that an `asreml()` call has been used to form the base model in step 1 before attempting QTL analysis. An error trapping function, `wgaim.default()` is called if the class of the base model is not `"asreml"`. The second argument `phenoData` is a data object of phenotypic data usually used in the analysis of the base model in step 1. The `intervalObj` contains the imputed genetic marker and interval data obtained from a call to `cross2int()` in the step 3. Thus `intervalObj` must be of class `"interval"`.

The character string `merge.by` is then used to identify the appropriate column of `phenoData` and `intervalObj` which to merge the two data sets. This merging differs depending on whether the problem is high dimensional, $(r \times m - c)$ or not. **Note: Unmatched elements of `merge.by` are handled differently depending on whether they are from the `intervalObj` or `phenoData`. If elements of `merge.by` exist in `phenoData` and are unmatched with elements in `intervalObj` then they are kept to ensure completeness of the phenotypic data. If elements of `merge.by` exist in `intervalObj` and not in `phenoData` they are dropped as there will be no phenotypic information available for that genetic line.**

The `gen.type` allows the user to specify `"interval"` or `"marker"` depending on the desired analysis. If `gen.type = "marker"` then the imputed marker matrix for each linkage group in `intervalObj` is combined into a whole genome matrix before being merged with `phenoData`. If `gen.type = "interval"` then the interval matrix for each linkage group is combined and used instead.

Two choices are available for the `method` argument. If `method = "fixed"` the forward selection algorithm moves the selected QTL to the fixed part of the model. This was the only choice in earlier versions of **wgaim** and is part of the original algorithm discussed in Verbyla et al. (2007). If `method = "random"` the forward selection algorithm uses the updated algorithm of Verbyla et al. (2012), also discussed briefly in Section 2.3, and places the selected QTL as an additive set of random effects.

The `selection` argument can either be `"chromosome"` or `"interval"`. If `"chromosome"` is chosen then selection of a QTL is based on outlier detection method discussed in Section 2 and in more detail in Verbyla et al. (2007). If `"interval"` is given then selection is

based on outlier interval statistics only. Either `selection` procedure can be used with both of choices of `method` discussed above. **Note: All combinations of the arguments discussed allow high dimensional genetic components to be added to the wgaim call through `intervalObj`.**

The `breakout` argument allows the user to breakout of the forward selection algorithm at the desired iteration by providing a positive integer. The default of -1 ensures the algorithm does not stop prematurely. `TypeI` argument allows users to change the significance level for the testing of QTL effects variance component γ_a. As `asreml()` calls output components of the fit to the screen there is an option to `trace` this to a file if desired. The level of reporting can be changed using `verboseLev`. If `verboseLev = 0` model fitting information and, if found, QTL locations will be printed. If `verboseLev = 1` then the chromosome (if necessary) and interval outlier statistics from (2.8) and (2.9) will be printed during each iteration.

Step 5. Summarise QTL with various method functions

```
summary(QTLmodel, intervalObj, LOD = TRUE, ...)
print(QTLmodel, intervalObj, ...)
tr(QTLmodel, iter = 1:length(object$QTL$effects), diag.out = TRUE, ...)
link.map(QTLmodel, intervalObj, chr, max.dist, marker.names = "markers",
        list.col = list(q.col = "light blue", m.col = "red", t.col =
        "light blue"), list.cex = list(t.cex = 0.6, m.cex = 0.6),
        trait.labels = NULL, tick = FALSE, ...)
```

Various functions can be used to summarize and diagnostically check the QTL obtained from a `wgaim()` analysis. The `summary()` function retrieves genetic marker information and assesses the significance of the QTL effects (fixed or random). For an interval analysis genetic information displayed includes chromosome and interval as well as name and location of flanking markers. For a marker analysis, chromosome, name and location of the closest linked marker are displayed. For both interval and marker analysis the size of the QTL effect, its significance and percent contribution to the genetic variance are also given. If `method = "fixed"` in the `wgaim` call then significance of the QTL effects are assessed from p-values calculated using Section 2.4.1. If `method = "random"` then probability values are calcualted using Section 2.4.2. LOD scores are also available for all QTL effects.

The `print()` method provides a simple annotated summary of the QTL as they were found during the `wgaim()` analysis.

`tr()` displays diagnostic information of the forward selection process underlying a `wgaim()` analysis. It shows a summary of the Residual Maximum Log-Likelihood ratio tests of

significance for the parameter γ_a at each iteration. There is also a triangular p-value or probability value matrix that shows the significance of the QTL effects at each iteration.

Selected QTL can also be placed on a linkage map using `link.map()`. This function neatly plots the linkage map and places `"interval"` or `"marker"` QTL at their appropriate position. The function has added flexibility for colouring of QTL regions as well as colour and size of printed text for all components of the map.

What is the best WGAIM analysis to use?

The different combination of the arguments, `gen.type`, `method` and `selection` in the `wgaim.asreml()` call produce 6 distinct WGAIM QTL analyses. The question and answers given below are to help guide users in choosing the appropriate combination of the arguments for the genotypic and phenotypic data they have. It should be noted that some of the answers provided are borne from gained knowledge and practical experience with the algorithm and software since its inception.

Q: I have a high dimensional linkage map.
A: The **wgaim** package has been updated to allow high dimensional maps to be incorporated and analysed efficiently for all combinations of the arguments.

Q: My linkage map contains several linkage groups that have small numbers of markers.
A: It is now known that using the using the chromosome outlier statistic wrongly favours selection of QTL from small linkage groups. It is advised to use `selection="interval"` in combination with the other arguments.

Q: My linkage map contains many linkage groups with sparsely spaced markers.
A: This would suggest the linkage map contains many wide intervals. It may be preferable to perform a marker analysis using `gen.type="marker"` in combination with the other arguments.

Q: My linkage map contains linkage groups with dense sets of markers.
A: With dense linkage maps QTL become tightly linked with markers. Therefore using either `gen.type="interval"` or `gen.type="marker"` will be efficient. The use of `selection="chromosome"` may also provide slight improvement in QTL selection.

Q: I am interested in the least biased QTL effects for a particular trait.
A: Using `method="random"` ensures the selected QTL will be placed as additive random components in the model. The QTL effects will therefore be shrunk and known to be less biased.

Q: It is suspected there are very closely linked QTL for a particular trait.
A: Very tightly linked QTL are difficult to determine and their simultaneous inclusion as separate covariates in any model may produce biased effects for one or both of the linked QTL. If these linked QTL are not of great interest users can adjust the `exclusion.window` argument to ensure that a cM region around each selected QTL is excluded from further analysis. If closely linked QTL are found using **wgaim** it may also be useful to post process the model by dropping each QTL independently and rechecking the results.

4 Package Examples

All results from the examples presented in this vignette are reproducible with data sets and scripts provided with the package. The scripts and vignette for your installation of **wgaim** and R can be found by typing the commands

```
R> docpath <- system.file("doc", package = "wgaim")
R> list.files(docpath)
```

The listed files in this directory should match

```
[1] "CxRExample.R"  "index.html" "RxKExample.R" "SxTExample.R" "wgaim.pdf"
```

If they do not match this or nothing is found then an upgrade of **wgaim** is needed. The newest version can be found at http://CRAN.R-project.org/package=wgaim. The data sets used in this vignette and available with the package are

```
R> data(package = "wgaim")

Data sets in package 'wgaim':

genoCxR                 Genotypic marker data for Cascades x RAC875-2
                        doubled haploid population in R/qtl format
genoRxK                 Genotypic marker data for RAC875 x Kukri doubled
                        haploid population in R/qtl format
genoSxT                 Genotypic marker data for Sunco x Tasman doubled
                        haploid population in R/qtl format
phenoCxR                Phenotypic Cascades x RAC875-2 zinc experiment
                        data
phenoRxK                Phenotypic RAC875 x Kukri trial data
phenoSxT                Phenotypic Sunco x Tasman trial data
```

They have been bundled with the package in two locations. Firstly, they are available in the "data" directory of the package and therefore can be locally retrieved using the usual

`data()` call. They have also been individually placed in an external data directory in CSV format. The path to this directory is locatable by printing the result of the following command

```
wgpath <- system.file("extdata", package = "wgaim")
```

Note that the three genotypic marker data sets in this directory are in raw CSV format.

4.1 RAC875 x Kukri data

This first example is used to illustrate the required steps for a successful **wgaim** analysis. It shows a more in depth view of the phenotypic and genotypic data and in particular focusses on the R/qtl linkage map, its conversion and use within **wgaim**.

The example consists of phenotypic and genotypic data sets involving a Doubled Haploid (DH) population derived from the interbreeding or crossing of the wheat varieties RAC875 and Kukri. The main goal of the experiment was to to find causal links between measured grain yield related traits and genetic markers associated with the population. The experiment was a subset of a much larger set of trials used for assessing drought tolerance of the breeding population across a variety of regions (see Bonneau et al., 2012; Bennett et al., 2012a,b,c).

The phenotypic RAC875 x Kukri data can be accessed using

```
R> data(phenoRxK, package = "wgaim")
```

and relates to a field trial consisting of 520 plots. Two replicates of 256 DH lines from the RAC875 x Kukri population were allocated to plots using a randomized complete block design with 2 Blocks/Reps. The additional plots remaining in each block were filled with one of each of the parents and controls (ATIL, SOKOLL, WEEBILL). A number of yield related trait measurements were taken and grain yield (t/ha) and thousand grain weight are included with this data.

The collected data frame consists of 520 Rows with 9 columns and an example of the first ten rows of data are given in Table 4.2. From left to right the "Genotype" column is a 256 level factor consisting of the unique identification of the DH lines, the parents and the controls. Type is a 4 level factor differentiating the DH lines from the parents and controls. "Row" and "Range" are 20 and 26 level factors determining the position of the experimental plot. "Rep" is a 2 level factor identifying the physical block each replicate of the DH lines was placed in. "yld" and "tgw" are the physical measurements of grain yield and thousand grain weight taken from each plot upon harvest. The final columns "lrow" and "lrange" is a centred numerical version of "Row" and "Range" that is used in

Table 4.1: The first 10 rows of the phenotypic RAC875 × Kukri experiment data

Genotype	Type	Row	Range	Rep	yld	tgw	lrow	lrange
DH_R003	DH	1	1	1	2.24	33.40	-12.50	-9.50
DH_R055	DH	2	1	1	1.16	31.60	-11.50	-9.50
DH_R056	DH	3	1	1	1.64	48.30	-10.50	-9.50
DH_R111	DH	4	1	1	2.40	31.60	-9.50	-9.50
DH_R112	DH	5	1	1	1.97	33.40	-8.50	-9.50
DH_R170	DH	6	1	1	1.27	26.30	-7.50	-9.50
DH_R172	DH	7	1	1	1.96	27.00	-6.50	-9.50
DH_R232	DH	8	1	1	2.17	28.40	-5.50	-9.50
DH_R234	DH	9	1	1	1.36	32.40	-4.50	-9.50
DH_R294	DH	10	1	1	1.07	29.40	-3.50	-9.50

the subsequent analysis.

4.1.1 Base Model

Initially, we begin with **Step 1** of the previous chapter by exploring a suitable base model for yield by considering (2.5) without the random regression effects, $\boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a}$, attributed to genetic markers/intervals, namely

```
R> rkyld.asi <- asreml(yld ~ Type, random = ~ Genotype + Rep,
+     rcov = ~ ar1(Range):ar1(Row), data = phenoRxK)
```

In the model, the `Genotype` variable is modelled as a set of polygenic random effects represented as $\boldsymbol{g}$ in (2.5). The `Rep` is included as a random effect represented by $\boldsymbol{u}_e$ (Smith et al., 2005, see). To ensure genetic differences between parental and progeny lines is captured the `Type` variable is modelled as a fixed effect, represented as $\boldsymbol{\tau}$ in (2.5). The residual error term, $\boldsymbol{e}$, of 2.5 is also modelled using the `rcov` argument of the asreml call. Typically, for a regular field trial of this type, a separable AR1 × AR1 process (AR1 = auto-regressive or order 1) is used to parametrically model correlation of the yield measurements existing due to adjacency of the plots in the field. A summary of the models variance parameter estimates shows a moderate correlation exists in the Range direction with a small correlation existing across the Rows.

```
R> summary(rkyld.asi)$varcomp
```

```
                          gamma   component   std.error    z.ratio     constraint
Genotype!Genotype.var 2.30479883 0.168047406 0.017093543  9.8310459       Positive
Rep!Rep.var           0.02371962 0.001729444 0.003916852  0.4415393       Positive
R!variance            1.00000000 0.072911963 0.007142522 10.2081538       Positive
R!Range.cor           0.24047738 0.240477376 0.068807980  3.4949053 Unconstrained
```

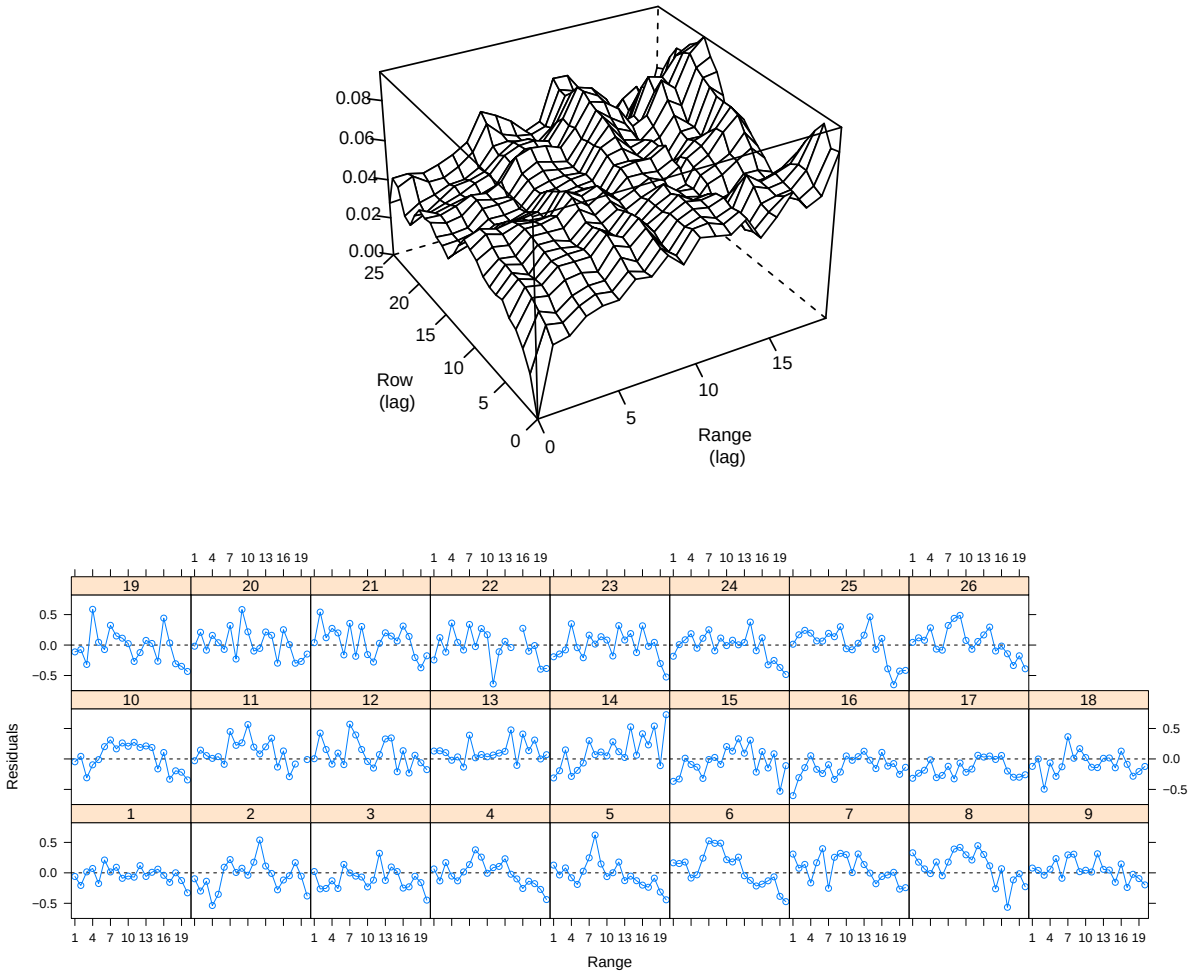

Figure 4.1: Row residuals for all Ranges from the initial model for yield in the RAC875 × Kukri experiment

```
R!Row.cor              0.50675409 0.506754095 0.048829027 10.3781321 Unconstrained
```

This initial model needs checking diagnostically. A simple `plot()` of the model object, which actually calls upon `plot.asreml()`, provides four diagnostic plots of the residuals for visual inspection. Trends across the field can be checked using the in-built **ASReml-R** variogram command or a simple trellis panel plot. For example, Figure 4.1 is produced with the two plotting commands

```
R> plot(variogram(rkyld.asi))
R> row.ind <- c(1,seq(4, 20, by = 3))
R> xyplot(resid(rkyld.asi) ~ Range | Row, data = phenoRxK, type = "b",
+     panel = function(x, y, ...){
+           panel.abline(h = 0, lty = 2)
+           panel.xyplot(x, y, ...)},
```

```
+     aspect = 4/5, layout = c(9,3), ylab = "Residuals",
+     scales = list(x = list(at = row.ind,labels = phenoRxK$Row[row.ind])))
```

The plots suggests there is a trend across the Rows for each of the Ranges. The variogram also shows a possible Range effect. Incorporating a linear row ("lrow") into the fixed effects and a "Range" random effect the full asreml model is

```
R> rkyld.asf <- asreml(yld ~ Type + lrow, random = ~ Genotype + Range,
+    rcov = ~ ar1(Range):ar1(Row), data = phenoRxK)
R> summary(rkyld.asf)$varcomp
```

```
                          gamma   component   std.error    z.ratio    constraint
Genotype!Genotype.var 3.11068122 0.165536298 0.016957534  9.7618141      Positive
Rep!Rep.var           0.06243523 0.003322519 0.005332801  0.6230345      Positive
Range!Range.var       0.29757224 0.015835440 0.006964592  2.2737068      Positive
R!variance            1.00000000 0.053215449 0.005157355 10.3183616      Positive
R!Range.cor           0.16334150 0.163341495 0.074785721  2.1841268 Unconstrained
R!Row.cor             0.26871404 0.268714042 0.072172541  3.7232172 Unconstarined
```

The summary suggests there is still a correlation in both the Row and Range direction after a linear de-trending across the Ranges. The addition of the spatial terms in the model has also reduced the residual variation without affecting the genetic variation. This has increased the heritability of the trait.

4.1.2 Genetic linkage map

We can now move to **Step 2** and read in a genetic marker map for the population. Similar to the phenotypic data, the RAC875 × Kukri genetic marker data is available using

```
R> data("genoRxK", package = "wgaim")
```

If the genotypic data is accessed in this manner the resultant object is a preformatted as an R/qtl `"cross"` object. Alternatively to illustrate the use of `read.cross()` in conjunction with **wgaim** the same data is available as a raw CSV file from the `extdata` directory of the package A subset of the data from the CSV file is given in Table 4.2. This reveals that the CSV file is in the rotated CSV format (see `read.cross()` from the **qtl** package). The genotypes are set as `AA` or `BB` and missing values are `"-"`. The consecutive missing values in the preview table are due to the combination of SSR and DaRT markers that were scored for different genotypes in the population before constructing the map. An appropriate call to `read.cross()` is

```
R> genoRxK <- read.cross("csvr", file="genoRxK.csv", genotypes=c("AA","BB"),
+    dir = wgpath, na.strings = c("-", "NA"))
R> class(raccas)
```

Table 4.2: Rotated CSV format of genetic linkage map for the RAC875 × Kukri population

Genotype			DH_R001	DH_R002	DH_R003	DH_R004	DH_R005	DH_R006
ksm0104a	1A	0.00	BB	AA	AA	BB	BB	BB
wPt-2527	1A	3.93	BB	AA	AA	BB	BB	BB
wPt-6564	1A	5.65	-	-	-	-	-	-
cfa2153	1A	5.88	BB	AA	AA	BB	BB	BB
wPt-7541	1A	6.79	-	-	-	-	-	-
wPt-6709	1A	6.79	-	-	-	-	-	-
gdm0033a	1A	8.05	BB	AA	AA	BB	BB	BB
wPt-6179	1A	9.08	-	-	-	-	-	-
wPt-8770	1A	9.08	BB	AA	AA	BB	-	BB

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

The returned object inherits the class `"bc"` (short for "back-cross"). If required, users can convert to a `"dh"` class by directly applying it to the object using the function `class()`. For the purpose of analysis and discussion in this report the two class types are synonymous and so the `"bc"` class is retained.

It is important to understand the elements of the R/qtl object before proceeding. Looking at the names of the object at the top level

```
R> names(genoRxK$geno)
```

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

In an R/qtl object, the `"pheno"` element is used to store the genotype names as well as hold other phenotypic information such as measured variables recorded for each genotype. In **wgaim** only the genotype names are used from `"pheno"` to assist in the merging of genotypic data with the external phenotypic data used to fit the base model described above.

```
R> genoRxK$pheno[["Genotype"]][1:18]
```

```
 [1] DH_R001 DH_R002 DH_R003 DH_R004 DH_R005 DH_R006 DH_R007 DH_R008 DH_R009
[10] DH_R010 DH_R011 DH_R012 DH_R013 DH_R014 DH_R015 DH_R016 DH_R017 DH_R018
```

A summary of the linkage map reveals there are 368 individuals genotyped with 500 markers spanning 21 linkage groups. Just over 10% of the marker scores are missing

```
R> summary(genoRxK)
```

```
    Backcross

    No. individuals:    368

    No. phenotypes:     1
    Percent phenotyped: 100

    No. chromosomes:    21
        Autosomes:      1A 1B 1D 2A 2B 2D 3A 3B 3D 4A 4B 4D 5A 5B 5D 6A 6B 6D 7A
                        7B 7D

    Total markers:      500
    No. markers:        44 37 26 22 23 16 24 57 21 22 19 6 12 19 5 21 32 8 47 21
                        18
    Percent genotyped:  89.5
    Genotypes (%):      AA:51.1  AB:48.9
```

Looking inside the "cross" object you will see the following

```
R> names(genoRxK$geno)

[1] "1A" "1B" "1D" "2A" "2B" "2D" "3A" "3B" "3D" "4A" "4B" "4D" "5A" "5B" "5D"
[16] "6A" "6B" "6D" "7A" "7B" "7D"
```

The genetic marker information is a named list format with the appropriate name for each linkage group. Looking deeper into the genetic object we see

```
R> names(genoRxK$geno$"3D")

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

For each linkage group, "data" contains the actual marker data matrix, converted into R/**qtl** format (AA = 1, BB = 2, missing values = NA). Marker names are placed as the column names. The rows of the data are in order of the genotype names found in genoRxK$pheno[["Genotype"]].

```
R> genoRxK$geno$"3D"$data[200:208,1:8]

     wPt-2464 cfd0079 cfd0064 cfd0034 wmc0533 wPt-6262 wPt-7894 barc0042
 [1,]        2       2       1       1       1       NA       NA        1
 [2,]        1       1       1       1       1        1        1        1
 [3,]        1       1       1       1       1        1       NA        1
 [4,]        1       1       1       1       1        2        2        2
 [5,]        2       2       1       1       1        1        1        1
```

```
[6,]    2    2    2    2    2    1    1    1
[7,]    2    1    1    1    1   NA   NA    1
[8,]    2    2    2    2    2    2    2    2
[9,]    1    1    2    2    1   NA   NA    1
```

The "map" element contains the map distances that may be either estimated using the Lander-Green hidden Markov algorithm (Lander & Green, 1987), or in this case, read in during the read.cross() process.

```
R> genoRxK$geno$"3D"$map
```

```
  wPt-2464    cfd0079    cfd0064    cfd0034    wmc0533   wPt-6262   wPt-7894
  0.000000   7.070536  53.420205  61.537557  70.659111  87.517197  94.406998
  barc0042    gwm0664   gwm0383b   gwm0314b   cfd0223b   barc0071   gwm0114a
108.448240 112.265676 116.367546 126.091701 134.965640 179.005013 181.480204
  wPt-5506    gwm0858   wPt-7241   wPt-2923   wPt-3412   barc0284   wPt-0485
181.480205 181.840812 182.607135 182.607135 182.607136 183.997849 186.021461
```

The first marker of the linkage group is always set to zero.

Following **Step 3** we now convert the "cross" object to an "interval" object. In doing so missing marker scores are imputed using the rules of Martinez & Curnow (1992) and consensus markers are created for co-located marker sets using the rules described in Table 3.1.

```
R> genoRxK <- cross2int(genoRxK, missgeno = "Mart", id = "Genotype",
+     rem.mark = TRUE)
R> class(genoRxK)
```

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

For this linkage map, a series of warning messages are outputted to the screen (omitted here) due to several lines containing a complete set of missing values for a linkage group. The missing values are replaced with zeros to ensure a complete linkage map (i.e. no missing values) is constructed. The classes of genoRxK and their ordering is retained and it now also inherits the class "interval" for use with functions in **wgaim**.

For a specific linkage group in the "interval" object, there are now additional components

```
R> names(genoRxK$geno$"3D")
```

```
[1] "data"         "map"          "dist"         "theta"        "imputed.data"
[6] "intval"
```

Thus for each linkage group, "data" and "map" are as before with the exception they now contain reduced sets of markers from omitting co-located markers. Markers proceeded by a "(C)" are now consensus markers

```
R> genoRxK$geno$"3D"$map
```

```
   wPt-2464     cfd0079     cfd0064     cfd0034     wmc0533    wPt-6262
   0.000000    7.070536   53.420205   61.537557   70.659111   87.517197
   wPt-7894     barc0042     gwm0664     gwm0383b    gwm0314b    cfd0223b
  94.406998  108.448240  112.265676  116.367546  126.091701  134.965640
   barc0071 gwm0114a(C)     gwm0858 wPt-7241(C)    barc0284    wPt-0485
 179.005013  181.480204  181.840812  182.607135  183.997849  186.021461
```

The additional components, "dist" contain the interval distances and "theta" are the recombination fractions between adjacent markers based on "dist". "imputed.data" contains the marker data with all missing values imputed

```
R> genoRxK$geno$"3D"$imputed.data[200:208,1:8]
```

```
        wPt-2464 cfd0079 cfd0064 cfd0034 wmc0533   wPt-6262   wPt-7894 barc0042
DH_R200       -1      -1       1       1       1  0.9333879  0.9370112        1
DH_R201        1       1       1       1       1  1.0000000  1.0000000        1
DH_R202        1       1       1       1       1  1.0000000  0.9809907        1
DH_R203        1       1       1       1       1 -1.0000000 -1.0000000       -1
DH_R204       -1      -1       1       1       1  1.0000000  1.0000000        1
DH_R205       -1      -1      -1      -1      -1  1.0000000  1.0000000        1
DH_R206       -1       1       1       1       1  0.9333879  0.9370112        1
DH_R207       -1      -1      -1      -1      -1 -1.0000000 -1.0000000       -1
DH_R208        1       1      -1      -1       1  0.9333879  0.9370112        1
```

and "intval" contains the interval data based on the mid-point pseudo-interval calculation of Verbyla et al. (2007) and defined as $\boldsymbol{M}_E$ in section 2.1.

```
R> genoRxK$geno$"3D"$intval[200:208,1:6]
```

```
           cfd0079    cfd0064    cfd0034    wmc0533   wPt-6262   wPt-7894
DH_R200 -0.9983369  0.0000000  0.9978094  0.9972358  0.9576392  0.9337226
DH_R201  0.9983369  0.9340516  0.9978094  0.9972358  0.9906333  0.9984207
DH_R202  0.9983369  0.9340516  0.9978094  0.9972358  0.9906333  0.9889310
DH_R203  0.9983369  0.9340516  0.9978094  0.9972358  0.0000000 -0.9984207
DH_R204 -0.9983369  0.0000000  0.9978094  0.9972358  0.9906333  0.9984207
```

```
DH_R205 -0.9983369 -0.9340516 -0.9978094 -0.9972358  0.0000000  0.9984207
DH_R206  0.0000000  0.9340516  0.9978094  0.9972358  0.9576392  0.9337226
DH_R207 -0.9983369 -0.9340516 -0.9978094 -0.9972358 -0.9906333 -0.9984207
DH_R208  0.9983369  0.0000000 -0.9978094  0.0000000  0.9576392  0.9337226
```

4.1.3 QTL analysis and summary

We have now have all the appropriate components of data to perform our wgaim QTL analysis in **Step 4**. For this analysis we will use the calculated genetic intervals or `"intval"` components of each linkage group and perform a fixed effects analysis, selecting QTL using interval statistics only. It is worthwhile understanding how `wgaim.asreml()` operates by breaking out of the forward selection algorithm after the first random effects interval model fit using the `breakout` argument

```
R> rkyld.qtl0 <- wgaim(rkyld.asf, phenoData = phenoRxK, intervalObj = genoRxK,
+     merge.by = "Genotype", trace = TRUE, na.method.X = "include",
+     gen.type = "interval", method = "fixed", selection = "interval",
+     breakout = 1, exclusion.window = 0)
```

In the initial hidden parts of this computation the phenotypic and genotypic interval data components are merged using the `merge.by` argument. For high dimensional genetic data a transformation is performed using section 2.3.2 and the details of Verbyla et al. (2012). This first model fit is then equivalent to (2.5) where all the intervals are included simultaneously into the linear mixed model with the extra term $\boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a}$. The BLUPs of the interval QTL effects are then recovered and the outlier statistics are formed to choose the first QTL. Both of these are returned with the object and can be found under `rkyld.qtl1$QTL$diag`. Figure 4.2 shows the scaled random interval QTL effects and the interval outlier statistics from the model from using the `out.stat()` function

```
R> out.stat(rkyld.qtl1, genoRxK, iter = 1, stat= "blups")
R> out.stat(rkyld.qtl1, genoRxK, iter = 1, stat= "os")
```

The plots highlight the linkage groups with separate colours and show the causal relationships the intervals have with yield. The first QTL, on chromosome 3B, that will be selected is also highlighted. A summary of the variance parameters of the model at this stage can be found using

```
R> asreml:::summary.asreml(rkyld.qtl1)$varcomp
```

```
                            gamma   component   std.error    z.ratio    constraint
ints!grp("ints").var  49.45607404 2.642258814 0.624502193  4.2309840      Positive
Genotype!Genotype.var  1.06306579 0.056795753 0.009362024  6.0666101      Positive
Rep!Rep.var            0.05987626 0.003198972 0.005172746  0.6184281      Positive
Range!Range.var        0.29750605 0.015894670 0.006964489  2.2822450      Positive
```

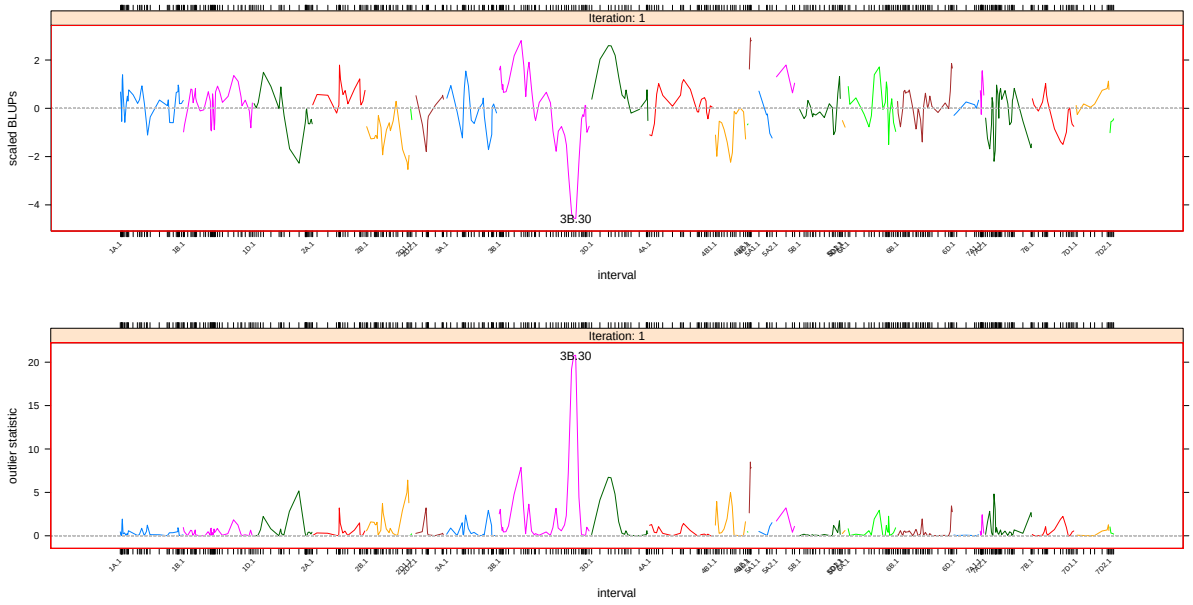

Figure 4.2: Scaled BLUPs of the interval QTL effects (TOP) and interval outlier statistics across the complete linkage map after the first model fit (BOTTOM).

```
R!variance          1.00000000 0.053426376 0.005133952 10.4064809      Positive
R!Range.cor         0.15120605 0.151206051 0.068950181  2.1929754 Unconstrained
R!Row.cor           0.29683957 0.296839574 0.066471843  4.4656438 Unconstrained
```

Comparing this to the variance parameter summary for `rkyld.asf` in section 4.1.1 an approximate percentage variance accounted for by the markers can be calculated as 100*(0.16553 - 0.0568)/0.16533 = 65.6%. This shortfall is not unusual for traits such as yield as they are known to be genetically complex.

Returning to the analysis the `breakout` argument is omitted from the `wgaim.asreml()` call (setting it back to default of -1) and the algorithm therefore continues until it halts.

```
R> rkyld.qtl1 <- wgaim(rkyld.asf, phenoData = phenoRxK, intervalObj = genoRxK,
+     merge.by = "Genotype", trace = TRUE, na.method.X = "include",
+     gen.type = "interval", method = "fixed", selection = "interval",
+     exclusion.window = 0)
```

By default, the tracing argument is `trace = TRUE` which produces an annotated version of the `asreml` models fitted throughout the forward selection algorithm. This output has been omitted for brevity. After the analysis is complete the QTL can be diagnostically checked and summarised using any of the method functions available in **Step 5**. In this example the summary of the resulting QTL is found using the method function `summary.wgaim()`, namely

```
R> summary(rkyld.qtl1, genoRxK, LOD = FALSE)
```

	Chromosome	Left Marker	dist(cM)	Right Marker	dist(cM)	Size	Pvalue	% Var
1	1D	wPt-1799	128.29	wPt-1263	166.85	-0.093	0.000	1.7
2	2A	wmc0296	84.77	wPt-7306	86.58	-0.205	0.003	8.4
3	2A	barc0220(C)	87.47	cfa2263	87.76	0.267	0.000	14.2
4	2B	wPt-9644	25.24	wPt-5672	29.97	-0.098	0.000	1.9
5	2B	wPt-3378	135.93	wPt-7360	136.11	-0.075	0.000	1.1
6	3B	wPt-7984	6.65	barc0075	7	0.073	0.000	1.0
7	3B	wmc0043	68.14	wPt-6973(C)	79.41	0.095	0.000	1.8
8	3B	wPt-8021	244.67	gwm0114b	256.42	-0.355	0.000	25.1
9	3B	gwm0114b	256.42	wPt-8845	265.8	0.181	0.000	6.5
10	3D	cfd0064	53.42	cfd0034	61.54	0.105	0.000	2.2
11	4D	gwm0297b	0	wmc0457	6.56	-0.195	0.006	7.6
12	4D	wmc0457	6.56	barc0288	7.32	0.293	0.000	17.2
13	7B	wPt-9925	93.88	wPt-5343	108.17	-0.063	0.003	0.8

The output for each QTL is summarised with the linkage group, the name and location of the flanking markers on the linkage group and the size of the QTL effect. The significance of the QTL effects are determined using the formula of section 2.4.1 and the percentage contribution to the overall genetic variance is calculated using section 2.4.3. Although Verbyla et al. (2007) recommends the use of p-values as the overall test of significance for each of the QTL, the argument `LOD = TRUE` can be given to `summary.wgaim()` if LOD scores are necessary. The analysis reveals 13 significant QTL across seven linkage groups. The summary also shows linkage groups 2A, 3B and 4D appear to have linked QTL in repulsion. Keen observers will realise the overall genetic contribution of the QTL is 89.5% which exceeds the original estimate of 65.6%. As section 3 indicates, this is most likely due to biased estimation of the individual genetic contributions of the tightly linked QTL. This phenomenon will be explored further in the next example.

4.2 Sunco x Tasman data

This example stresses the importance of modelling extraneous variation to a ensure a more appropriate QTL analysis. It is also used to highlight the diagnostic and visual features of **wgaim**. The Sunco x Tasman data sets consist of phenotypic milling trial data as well as a genetic linkage map involving a doubled haploid population formed from the crossing of wheat varieties Sunco x Tasman. The aim of the experiment was to determine genetic markers that may be linked to milling yield.

The phenotypic data can be accessed using

```
R> data(phenoSxT, package = "wgaim")
```

The data relates to a two phase experiment involving 175 DH lines of Sunco x Tasman, 2 parents and 6 commercial lines. The first phase of the experiment was a field trial conducted in the year 2000 consisting of 31 rows and 12 columns. DH lines were then allocated to plots in this spatial array using a randomized complete block design with 2 Blocks. Additional plots were filled with the parents and commercial lines. A second phase milling experiment was then carried out where 23% of the field plots were replicated to produce a total of 456 milling samples. These partially replicated field samples were then randomly allocated to 38 mill days with 12 milling samples per day. The focus is on the trait milling yield.

The data frame consists of 456 rows with 12 columns

```
R> names(phenoSxT)
```

```
 [1] "Expt"    "Type"    "id"      "Range"   "Row"     "Rep"     "Millday"
 [8] "Millord" "myield"  "lord"    "lrow"
```

In this example "Type" is a 9 level factor distinguishing the DH, parents and commercial lines. The "id" columns is a 183 level factor containing a unique identification of the 175 DH line and 8 other wheat varieties. The original field Row and Range (Column) have been kept and are numeric factors of 31 and 12 levels respectively. "Rep" represents the two level Blocking structure from the field. Similarly, "Millday" and "Millord" are numeric factors of 38 and 12 levels respectively arising from the milling design. "myield" is a quantitative variable capturing the milling yield of each of the samples. The final two variables are centred quantitative equivalents of the factors Row and Millord.

4.2.1 Base model

It is important to understand the hierarchical structure of data arising from a two phase experiment prior to statistical modelling. Smith et al. (2006) provides and excellent initial reference and in particular the ANOVA table of a hypothetical milling experiment in Table 5 of this article shows terms appropriate for inclusion in an initial model. Using this table as a guide an appropriate initial model would be

```
R> st.fmI <- asreml(myield ~ Type, random = ~ id + Rep + Range:Row +
+     Millday, rcov = ~ Millday:ar1(Millord), data = phenoSxT)
```

Due to the natural hierarchy existing in the data, diagnostically, there are several components of this model that need checking. The (milling) residuals of the model can be checked with

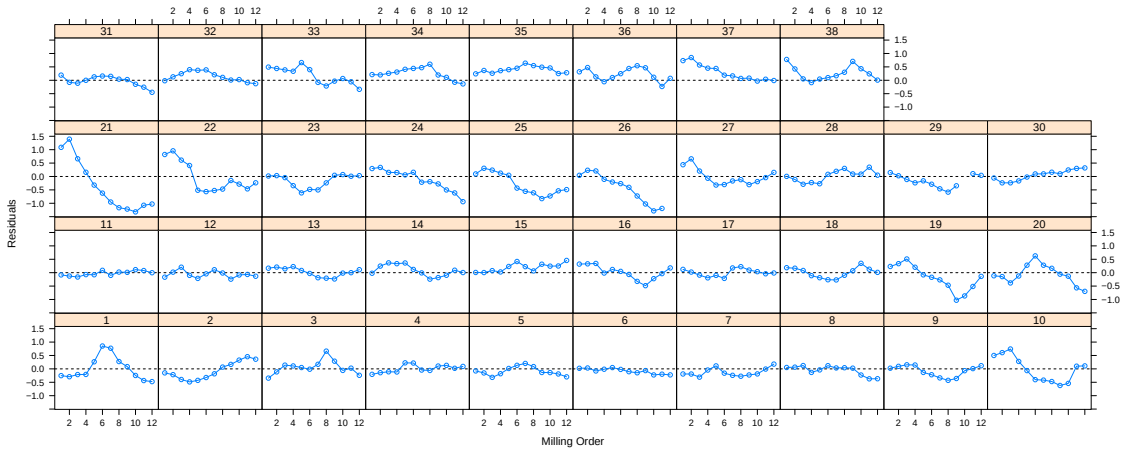

Figure 4.3: Milling residuals from initial model of Sunco x Tasman milling experiment

```
R> xyplot(resid(st.fmI) ~ as.numeric(Millord) | Millday, data =
+     phenoSxT, type = "b")
```

and are given in Figure 4.3. The plot suggest there may be a slight downward trend in milling yield across the order of milling samples each day. In a similar manner the field residuals can also be plotted by extracting the random effect coefficients from the "Range:Row" term of the model.

```
R> field.resid <- coef(st.fmI, pattern = "Range:Row")
R> rrd <- data.frame(field.resid = field.resid,
+     Range = factor(rep(1:12, each = 31)), Row = factor(rep(1:31,12)))
R> xyplot(field.resid ~ Row | Range, data = rrd, type = "b", layout = c(6,2))
```

Figure 4.4 shows the field residuals across Rows for given Ranges and indicates there is slight donward trend in milling yield across the Rows of the field.

To compensate for these trends a linear row ("lrow") and linear order ("lord") terms are fitted as fixed effects in the asreml model. Thus the full base asreml model is of the form

```
R> st.fmF <- asreml(myield ~ Type + lord + lrow, random = ~ id + Rep +
+     Range:Row + Millday, rcov = ~ Millday:ar1(Millord), data = phenoSxT)
R> summary(st.fmF)$varcomp
```

```
                 gamma  component  std.error   z.ratio    constraint
id           7.0925458 1.92573995 0.23965934 8.0353220      Positive
Rep          0.2843737 0.07721201 0.15604795 0.4947967      Positive
Range:Row    1.4973306 0.40654927 0.06206771 6.5500926      Positive
```

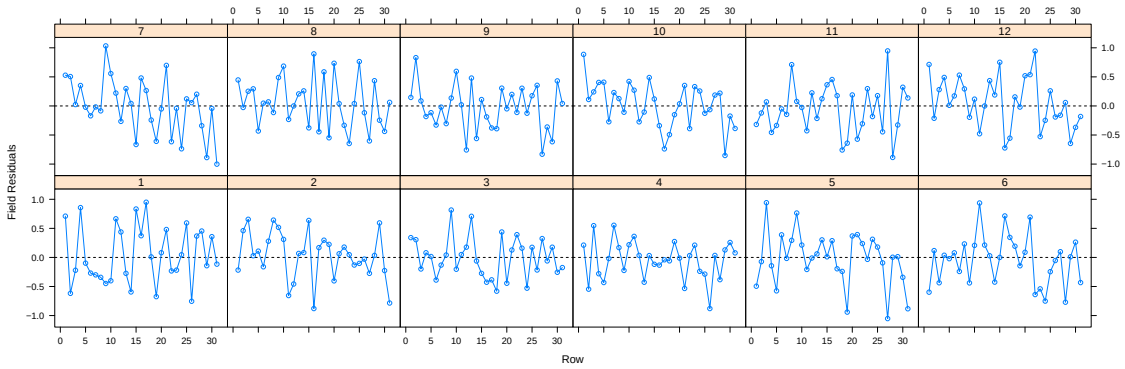

Figure 4.4: Field residuals from initial model of Sunco x Tasman milling experiment

```
Millday       1.7795039 0.48316385 0.15646257 3.0880476      Positive
R!variance    1.0000000 0.27151604 0.08035809 3.3788264      Positive
R!Millord.cor 0.7109431 0.71094307 0.12682697 5.6056142 Unconstrained
```

The summary reveals a large genetic variance component. For comparison a NULL model (no extraneous effects) is also fitted.

```
R> st.fmN <- asreml(myield ~ 1, random = ~ id, data = phenoSxT,
+     na.method.X = "include")
```

4.2.2 Linkage map

The genetic linkage map for the Sunco x Tasman population can be loaded using either of the first two following commands

```
R> genoSxT <- data(genoSxT, package = "wgaim")
R> genoSxT <- read.cross("csv", file="genoSxT.csv", genotypes=c("AA","BB"),
+     dir = wgpath, na.strings = c("-", "NA"))
R> nmar(genoSxT)
```

```
1A 1B 1D 2A 2B 2D 3A 3B 3D 4A 4B 4D 5A 5B 5D 6A 6B 6D 7A 7B 7D
 9 16 15 13 12 22 12 16 13 19 13  8 18 17  8  5 16  6 31 13  5
```

The map consists of 190 individuals that have been genotyped with 287 markers. After some exploration of the linkage map there appears to be some individuals that have less than half of their markers scored. The individuals do not feature in the phenotypic data set and therefore can be safely discarded.

```
R> nt <- ntyped(genoSxT, "ind")
R> nt[nt < 120]
```

```
186 187 188 189 190 191 192 194 195 196
 75  73  72  64  71  74  67 112 110 108

R> genoSxT <- subset(genoSxT, ind = 1:180)
R> genoSxT <- cross2int(genoSxT, missgeno="Mart", id = "id" rem.mark = FALSE)
```

After omitting the non-essential lines the linkage map is converted from a `"cross"` object to an `"interval"` object. The original map did not contain co-located markers

It is possible to view the genetic map using `link.map()` in various ways. The function allows sub-setting according to distance (cM) and/or chromosome. Figure 4.5 shows two maps with the top one representing the all 21 linkage groups with no subsetting. The bottom map is subsetted by using the `"chr.dist"` argument which takes either or both `start` and `end` elements. These elements can have a single distance (cM) or a vector of distances matching the number of chromosomes from `"chr"`.

```
R> link.map(genoSxT, chr = names(nmar(genoSxT)), m.cex = 0.5)
R> link.map(genoSxT, names(nmar(genoSxT)[1:20]), m.cex = 0.5,
+      chr.dist = list(start = 25, end = 180), marker.names = "dist")
```

For larger maps a more aesthetic plot is reached by adjusting the marker character expansion (`m.cex`) parameter and increasing the plotting window width manually.

4.2.3 QTL analysis and diagnostics

A QTL analysis is now performed for the full model `st.fmF` and the null model `st.fmN`. This time we pipe the non-essential output to a text file using a file name for the argument `trace`. After doing this, only individual QTL found are annotated to the screen (omitted here).

```
R> st.qtlN <- wgaim(st.fmN, phenoData = phenoSxT, intervalObj = genoSxT,
+     merge.by = "id", gen.type = "interval", method = "fixed",
+     selection = "interval", trace = "nullmodel.txt",
+     exclusion.window = 0)
R> st.qtlF <- wgaim(st.fmF, phenoData = phenoSxT, intervalObj = genoSxT,
+     merge.by = "id", gen.type = "interval", method = "fixed",
+     selection = "interval", trace = "fullmodel.txt",
+     exclusion.window = 0)
```

In a similar fashion to the last example, the process of selecting QTL is determined from the outlier statistics. These are saved, along with the BLUP interval effects, for each iteration and can be viewed using the `out.stat()` command. For the first two iterations of the process the BLUP interval effects and interval outliers statistics are given in Figure

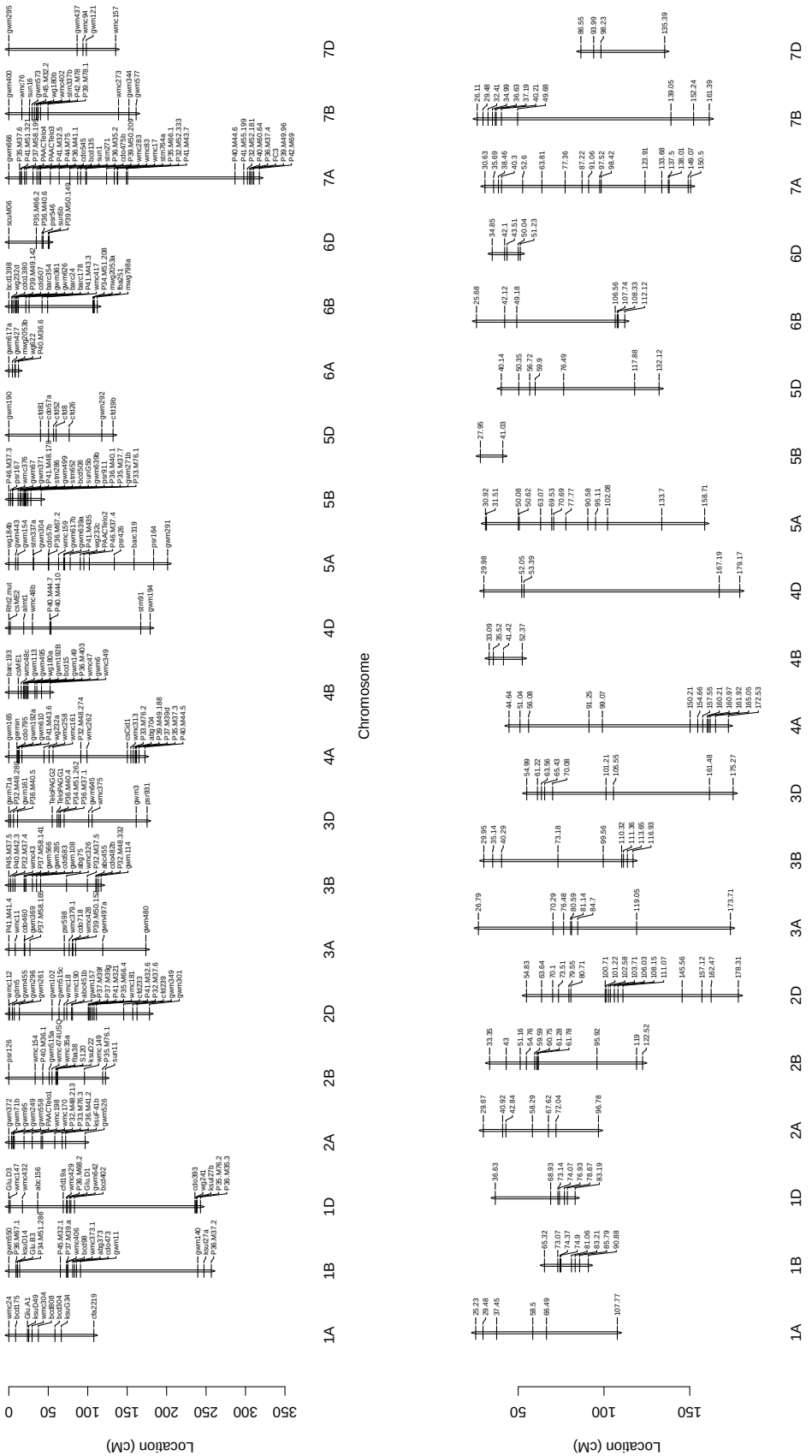

Figure 4.5: The linkage groups of the genetic map for the Sunco-Tasman data. Names of chromosomes are given at the bottom and genetic distances between markers are placed alongside each of the chromosomes.

4.6 are produced with

```
R> out.stat(st.qtlF, genoSxT, int = TRUE, iter = 1:2, cex = 0.6, stat = "os")
R> out.stat(st.qtlF, genoSxT, int = TRUE, iter = 1:2, cex = 0.6, stat = "blups")
```

The plots show the causal links between the interval QTL and milling yield. For larger or denser linkage maps there is also an additional argument that allows the user to subset the map to specific chromosomes which is only available when `int = TRUE`. (Figure omitted). For this example, the plots highlight the large QTL existing on 2B and 6B and also show the wide QTL existing on 1B.

```
R> out.stat(st.qtlF, genoSxT, int = TRUE, iter = 1:5, cex = 0.6,
+     chr = c("2B","4B","6B","7D"))
```

From a statistical standpoint the QTL selected across the genome cannot be expected to be orthogonal. Thus the introduction of the next QTL in the forward selection process will inevitably affect the significance of the previously selected QTL. A post diagnostic evaluation of the QTL p-values in the forward selection process can be displayed using

```
R> tr(st.qtlF, iter = 1:10, digits = 3)

Incremental QTL P-value Matrix.
================================
          2B.5   6B.5   7D.2   4B.1  1B.13   4D.1  5A.13  2A.7  3D.5  1B.4
Iter.1  <0.001
Iter.2  <0.001 <0.001
Iter.3  <0.001 <0.001 <0.001
Iter.4  <0.001 <0.001 <0.001 <0.001
Iter.5  <0.001 <0.001 <0.001 <0.001  0.001
Iter.6  <0.001 <0.001 <0.001 <0.001 <0.001 <0.001
Iter.7  <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 <0.001
Iter.8  <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 0.006
Iter.9  <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 0.005 0.012
Iter.10 <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 0.003 0.009 0.013

Likelihood Ratio Test of QTL Variance Component.
=========================================
              L0       L1 Statistic Pvalue
Iter.1  -309.563 -251.036   117.054 <0.001
Iter.2  -279.819 -243.841    71.955 <0.001
Iter.3  -269.714 -239.729     59.97 <0.001
Iter.4  -262.115 -236.919    50.392 <0.001
Iter.5  -247.277 -233.758    27.038 <0.001
Iter.6  -241.707 -230.061    23.293 <0.001
```

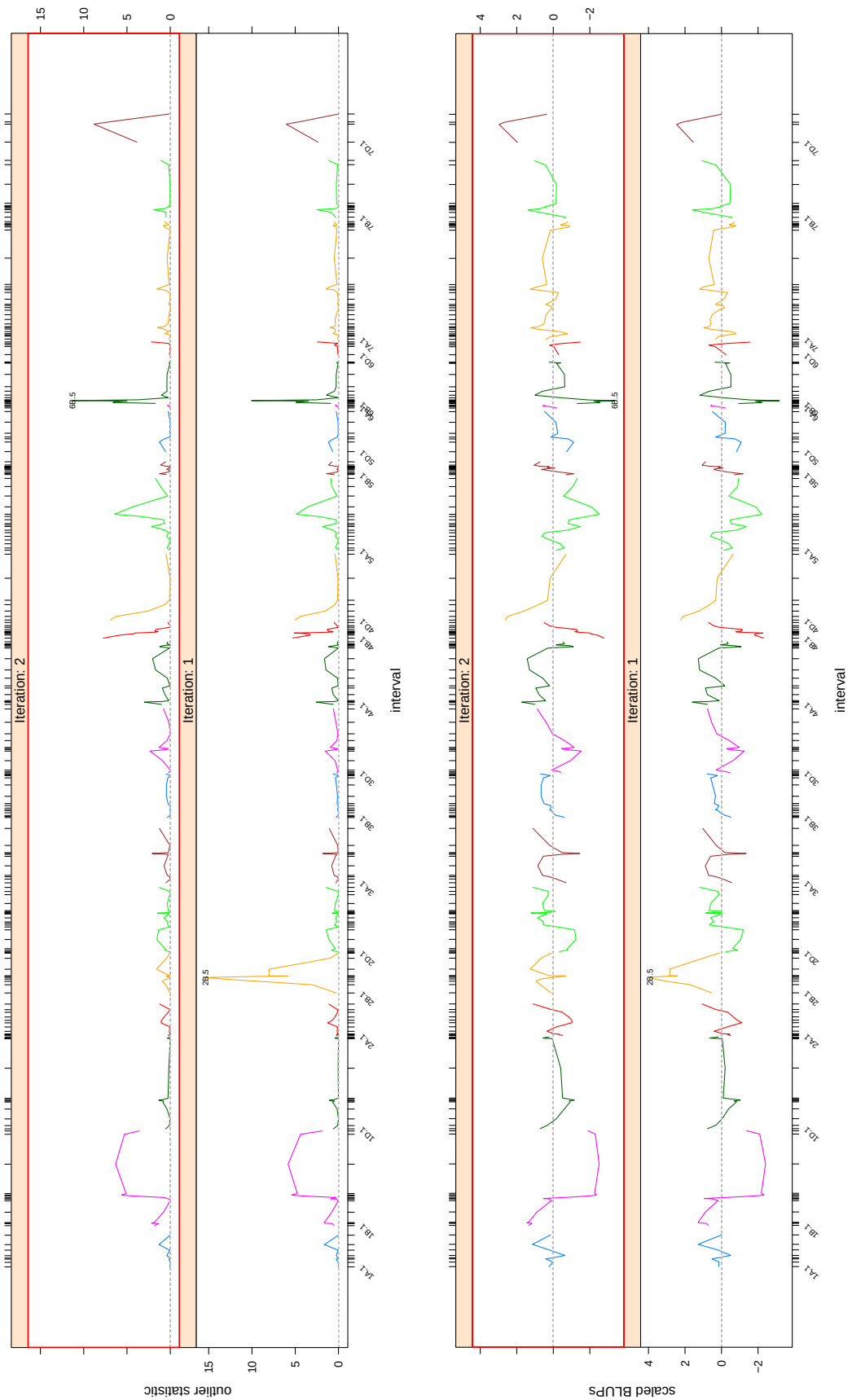

Figure 4.6: BLUP interval effects and interval outlier statistics for the first two iterations of the wgaim fit for the full model.

```
Iter.7  -236.851  -226.46    20.782 <0.001
Iter.8   -226.72 -222.799     7.842  0.003
Iter.9  -223.931  -221.89     4.081  0.022
Iter.10 -223.504 -221.524      3.96  0.023
Iter.11 -223.045 -221.349     3.391  0.033
Iter.12 -223.089 -221.038     4.102  0.021
Iter.13 -223.033 -221.352     3.363  0.033
Iter.14 -223.089 -221.659      2.86  0.045
Iter.15 -222.718 -221.791     1.854  0.087
```

The first of these displays shows the p-values of the selected QTL for the first ten iterations occurring in the WGAIM process. An example of the dynamic changes in significance can be seen for the selected QTL interval 2A.7. The second display presents the likelihood ratio tests, $-2\log\mathbf{\Lambda}$, for the significance of the QTL variance parameter, γ_a, in (2.5), with the inclusion of the last hypothesis test where the null model is retained. Both of these diagnostics are useful in determining the strength of the putative QTL entering the fixed model and the effects it has on QTL already selected.

4.2.4 Visualising your QTL results

Similar to the previous example, full summaries are available through the `summary.wgaim()` command. From an interval analysis complete information on each QTL is provided including names and distances of the flanking markers as well size, signficance and the contribution of the QTL to the overall genetic variance.

```
R> summary(st.qtlF, gneoSxT, LOD = FALSE)
```

	Chromosome	Left Marker	dist(cM)	Right Marker	dist(cM)	Size	Pvalue	% Var
1	1B	Glu.B3	11.02	P34.M51.286	13.71	0.196	0.003	1.4
2	1B	gwm11	90.88	gwm140	239.51	-0.812	0.000	23.4
3	2A	wmc198	29.67	wmc170	40.92	-0.225	0.002	1.8
4	2B	wmc474USQ	54.76	wmc35a	59.59	0.840	0.000	25.1
5	3D	TeloPAGG2	54.99	TeloPAGG1	61.22	-0.215	0.002	1.6
6	4A	germin	10.32	cdo795	11.39	0.153	0.021	0.8
7	4B	barc193	0	csME1	11.98	-0.445	0.000	7.0
8	4D	Rht2.mut	0	csME2	1.84	0.309	0.000	3.4
9	5A	wmc159	63.07	gwm617b	69.53	-0.181	0.025	1.2
10	5A	PAACTelo2	95.11	P46.M37.4	102.08	-0.273	0.001	2.6
11	5D	cfd81	40.14	cdo57a	50.35	-0.145	0.029	0.7
12	6B	cdo507	8.92	barc354	9.45	-0.644	0.000	14.7
13	6B	barc24	21.58	barc178	25.68	0.252	0.009	2.3
14	7D	gwm437	86.55	wmc94	93.99	0.305	0.000	3.3

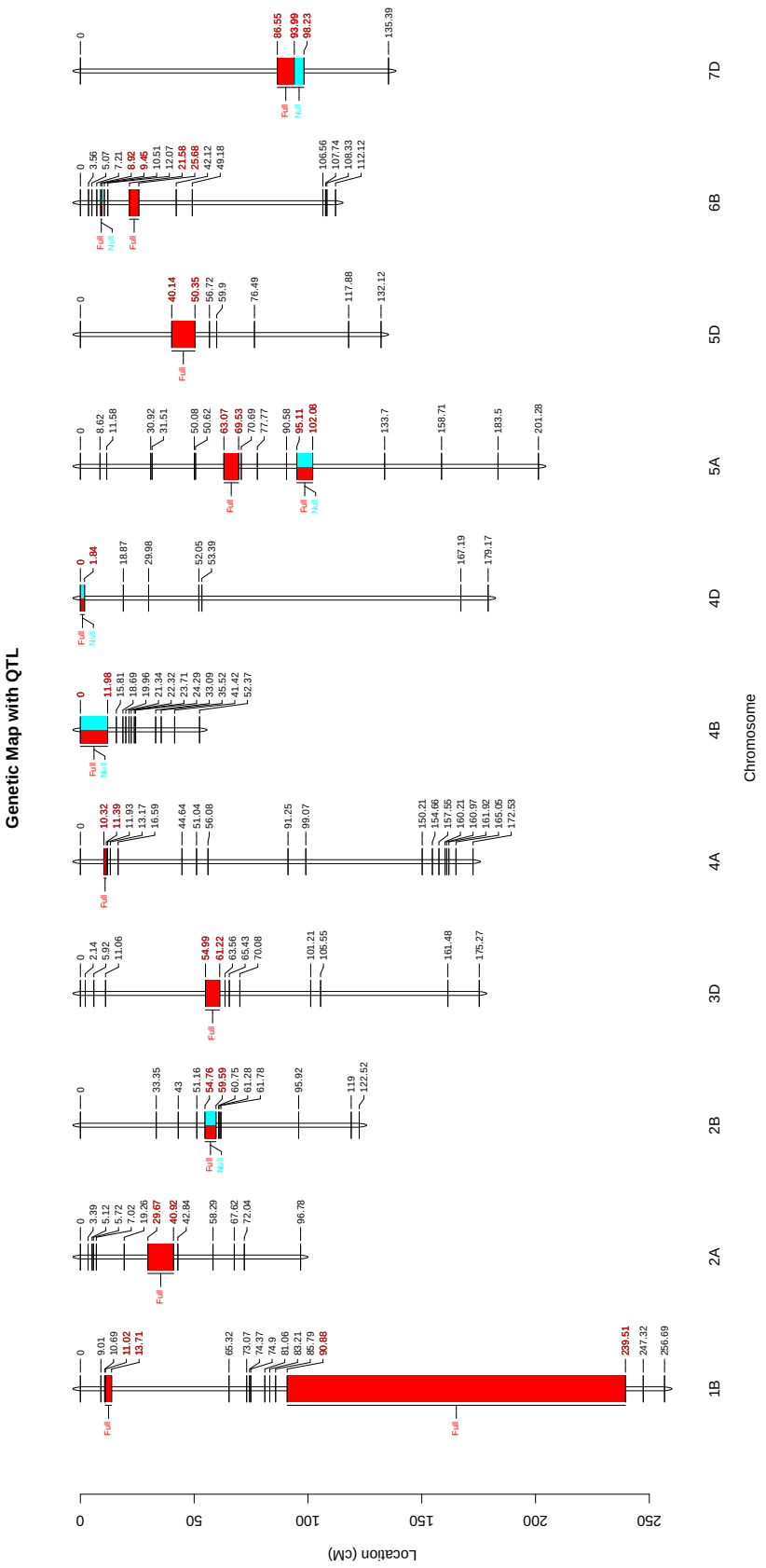

Figure 4.7: Genetic map with QTL for the Full and Null models obtained from an analysis of the Sunco x Tasman data. Markers and intervals for the QTL are highlighted and trait names are placed on the left hand side of the chromosomes.

The summary shows some large milling yield QTL have been found on several chromosomes. Of particular note is the large QTL on 6B in a very small interval of 0.5cM. In contrast, a large QTL was found on chromosome 1B in a 100+cM interval. There are also tightly linked QTL found on 6B. These QTL wll be explored in more detail in section 4.2.6. The summary produces a `data.frame` of results that can be easily exported to a spreadsheet program if desired. For multiple tables a simple table binding function is provided which stacks the QTL tables making it instantly useful for exporting with programs such as the the LaTeX table package **xtable**. (table omitted here.)

```
R> qtlTable(st.qtlF, st.qtlN, intervalObj = genoSxT, labels = c("Full",
+     "Null"), columns = 1:8)
```

The full and the NULL QTL models can be summarised visually using `link.map()`. In this case it calls the method `link.map.wgaim()` to plot the QTL on the genetic map.

```
R> link.map(st.qtlF, genoSxT, marker.names = "dist", cex = 0.6,
+     trait.labels = "Full")
```

Multiple models or traits can be handled through `link.map.default()`. For example, Figure 4.7 is produced with

```
R> link.map.default(list(st.qtlF, st.qtlN), genoSxT, marker.names = "dist",
+     trait.labels = c("Full", "Null"))
```

The multiple QTL map reveals that an extra six QTL were detected in the full model compared to the null model, highlighting the importance of modelling extraneous variation appropriately in QTL analyses.

The QTL plotting procedures `link.map.wgaim()` and `link.map.default()` are highly customisable. Through an argument `list.col` it allows the user to specify the QTL colour between markers, the colour of the flanking QTL marker names, the colour of the trait names and the rest of the marker names. If no colours are chosen `q.col` and `t.col` defaults to `rainbow(n)` where n is the number of traits. You can also change the size of the marker and trait name text with the argument `list.cex`.

Some customized examples are given below for the Full and Null QTL models for the Sunco x Tasman data and can be seen in Figure 4.8. These have been produced using the following criteria; change the colour of the QTL regions and the names and setting the background marker text grey.

```
R> link.map.default(list(st.qtlF, st.qtlN), genoSxT, marker.names = "dist",
+     trait.labels = c("Full", "Null"), list.col = list(q.col = c("skyblue3",
+     "salmon2"), m.col = "red", t.col = c("skyblue3", "salmon2")), col = "gray")
```

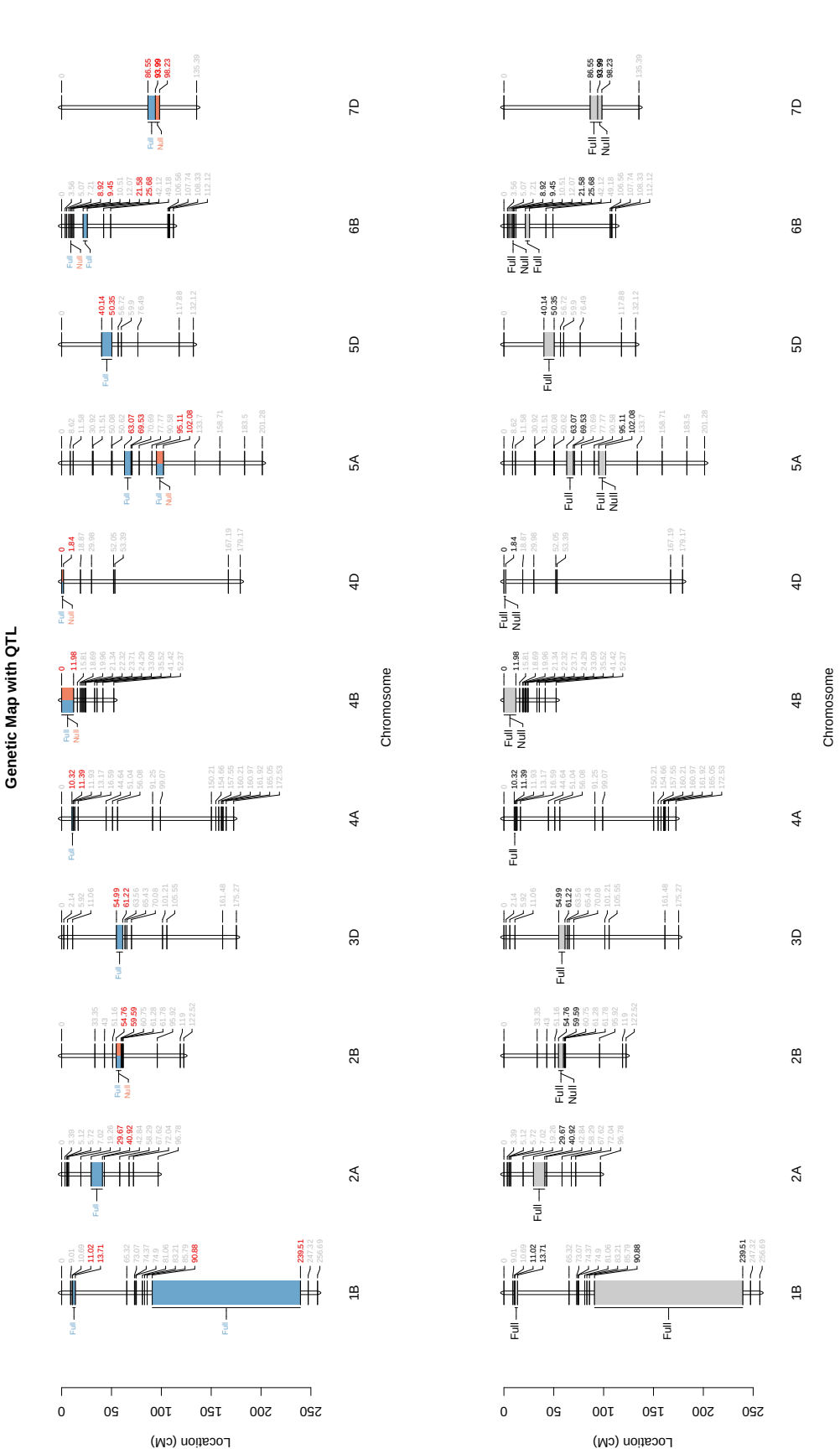

Figure 4.8: Customized genetic maps with QTL for the full and Null models obtained from an analysis of the Sunco x Tasman data.

A monochromatic plot with increased sizes for the trait labels.

```
R> link.map.default(list(st.qtlF, st.qtlN), genoSxT, marker.names = "dist",
+     trait.labels = c("Full", "Null"), list.col = list(q.col =
+     rep(gray(0.8), 2), m.col = "black", t.col = "black"),
+     list.cex = list(t.cex = 0.8), col = "gray")
```

4.2.5 Marker analysis

The summary of the interval QTL analysis for the full model shows a putative QTL in a very large interval on 1B. It may then be of interest to perform a marker analysis to see if this QTL is more closely linked to either of the flanking markers. This can be done by simply changing the `gen.type` argument in the call

```
R> st.qtlFM <- wgaim(st.fmF, phenoData = phenoSxT, intervalObj = genoSxT,
+     merge.by = "id", gen.type = "marker", method = "fixed",
+     selection = "interval", trace = "fullmodel.txt", exclsuion.window = 0)
```

The `wgaim` model can be summarised in the usual way.

```
R> summary(st.qtlFM, genoSxT, LOD = TRUE)
```

```
   Chromosome      Marker dist(cM)   Size Pvalue % Var    LOD
1          1B P34.M51.286    13.71  0.203  0.002   1.1  2.135
2          1B      cdo473    85.79 -0.304  0.000   2.4  4.414
3          1B     ksuI27a   247.32 -0.230  0.000   1.4  3.097
4          2A      wmc198    29.67 -0.180  0.008   0.8  1.504
5          2B   wmc474USQ    54.76  0.774  0.000  15.6 26.492
6          3D   TeloPAGG2    54.99 -0.190  0.004   0.9  1.785
7          4A      germin    10.32  0.176  0.007   0.8  1.584
8          4B       csME1    11.98 -0.411  0.000   4.4  7.274
9          4D    Rht2.mut        0  0.287  0.000   2.1  4.652
10         5A   P46.M37.4   102.08 -0.334  0.000   2.9  5.435
11         6B     barc354     9.45 -1.244  0.000  40.4  4.472
12         6B      gwm361    10.51  0.810  0.003  17.1  1.897
13         7D       wmc94    93.99  0.289  0.000   2.2  4.672
```

For marker QTL analysis the summary output is identical to the output of the interval QTL analysis with the exception that only the closest linked marker name and location is given for each QTL. The summary shows the large QTL found on 1B in the interval analysis has been reduced to two small QTL linked to the flanking markers. This indicates there is most likely large QTL existing in the wide interval. The summary also shows a large QTL on 2B in exactly the same region as the QTL found using interval analysis.

The marker analysis also found 2 QTL on 6B in close proximity that soak up a sizeable portion of the genetic variation.

The outlier statistics for this analysis are at the marker positions and can also be plotted using `out.stat`. The marker outlier statistics for the first first five iterations can be seen in Figure 4.9

```
R> out.stat(st.qtlFM, genoSxT, int = TRUE, iter = 1:5, cex = 0.6)
```

Fitting the Null model in a similar manner.

```
R> st.qtlNM <- wgaim(st.fmN, phenoData = phenoSxT, intervalObj = genoSxT,
+     merge.by = "id", gen.type = "marker", method = "fixed",
+     selection = "interval", trace = "nullmodel.txt", exclusion.window = 0)
```

Similar to the interval analysis, the results from the Full model and the Null model can be plotted on the linkage map and is given in Figure 4.10. The QTL are now highlighted with plotting symbols that can be altered with the usual arguments, `pch` and `cex`.

```
R> link.map.default(list(st.qtlFM, st.qtlNM), genoSxT, marker.names = "dist",
+     trait.labels = c("Full", "Null"), list.col = list(q.col = c("red",
+     "light blue"), m.col = "red", t.col = c("red", "light blue")),
+     list.cex = list(t.cex = 0.9, m.cex = 0.7), col = "black",
+     cex = 2, pch = 16)
```

Again, the plot reveals that the Null model discovered less QTL than the Full model.

4.2.6 Exclusion window

Both the interval analysis and the marker analysis of the full model indicate there were two tightly linked QTL in repulsion selected on chromosome 6B. Checking the scaled BLUPs from the interval analysis the reason for the selections are revealed.

```
R> out.stat(st.qtlF, genoSxT, iter = c(2,3,11), cex = 0.6,
+     chr = c("6B","7D"), stat = "blups")
```

After the selection of the first QTL on 6B in iteration 2 and its subsequent fixed effects estimation, the BLUPs in the proximity of the chosen QTL appear to change sign from negative to positive. This is not unusual and indicates that the first QTL was hiding another tightly linked QTL of opposite effect. This QTL is eventually chosen in iteration 11 of the algorithm. Unfortunately very tightly linked QTL have minimal recombination between them, indicating that the selection of the second QTL is heavily based on the phenotypic information stemming from the small number of lines that have recombined

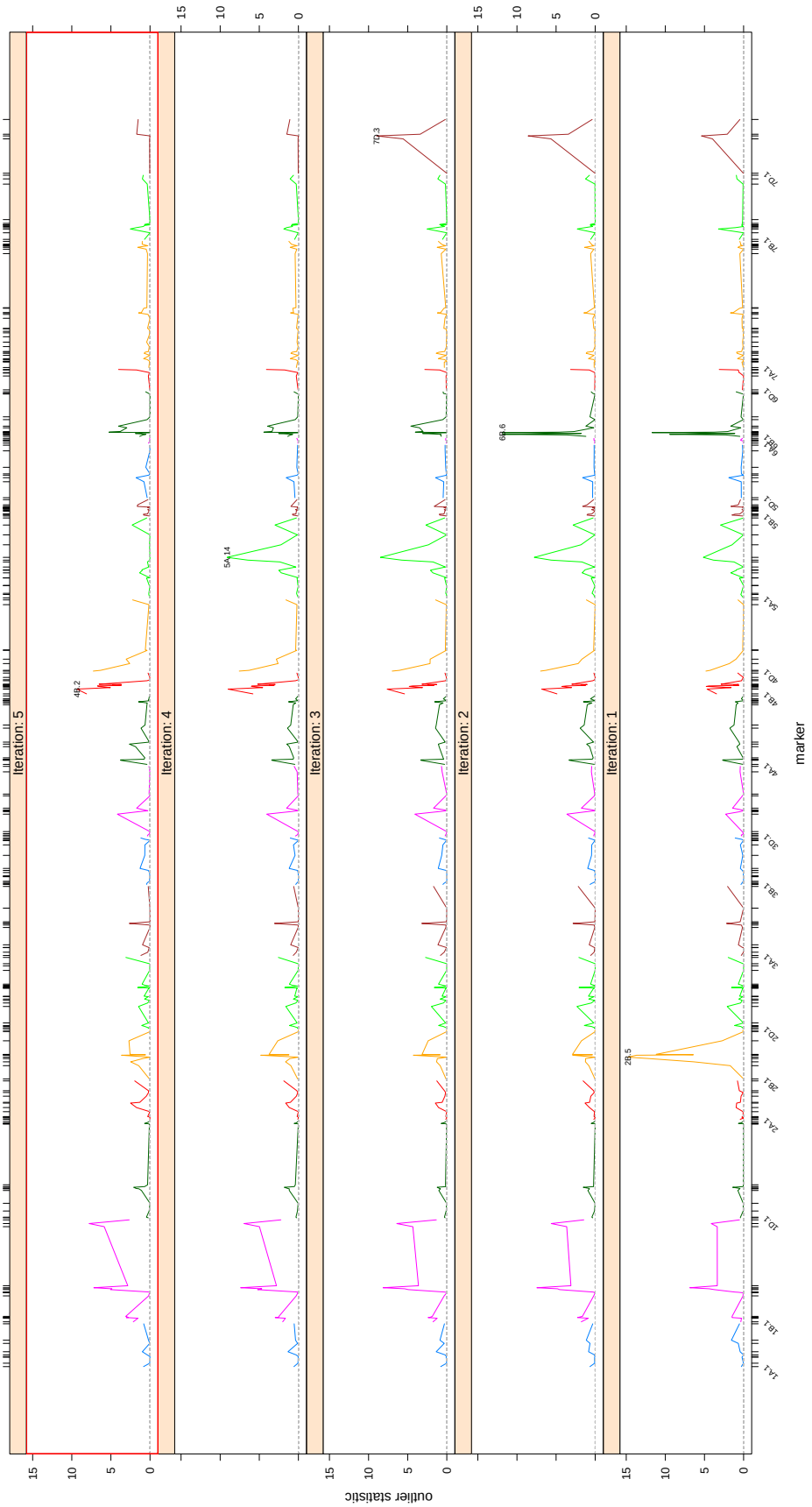

Figure 4.9: Outlier statistic plots for the first five iterations of the full marker regression model using the random WGAIM method

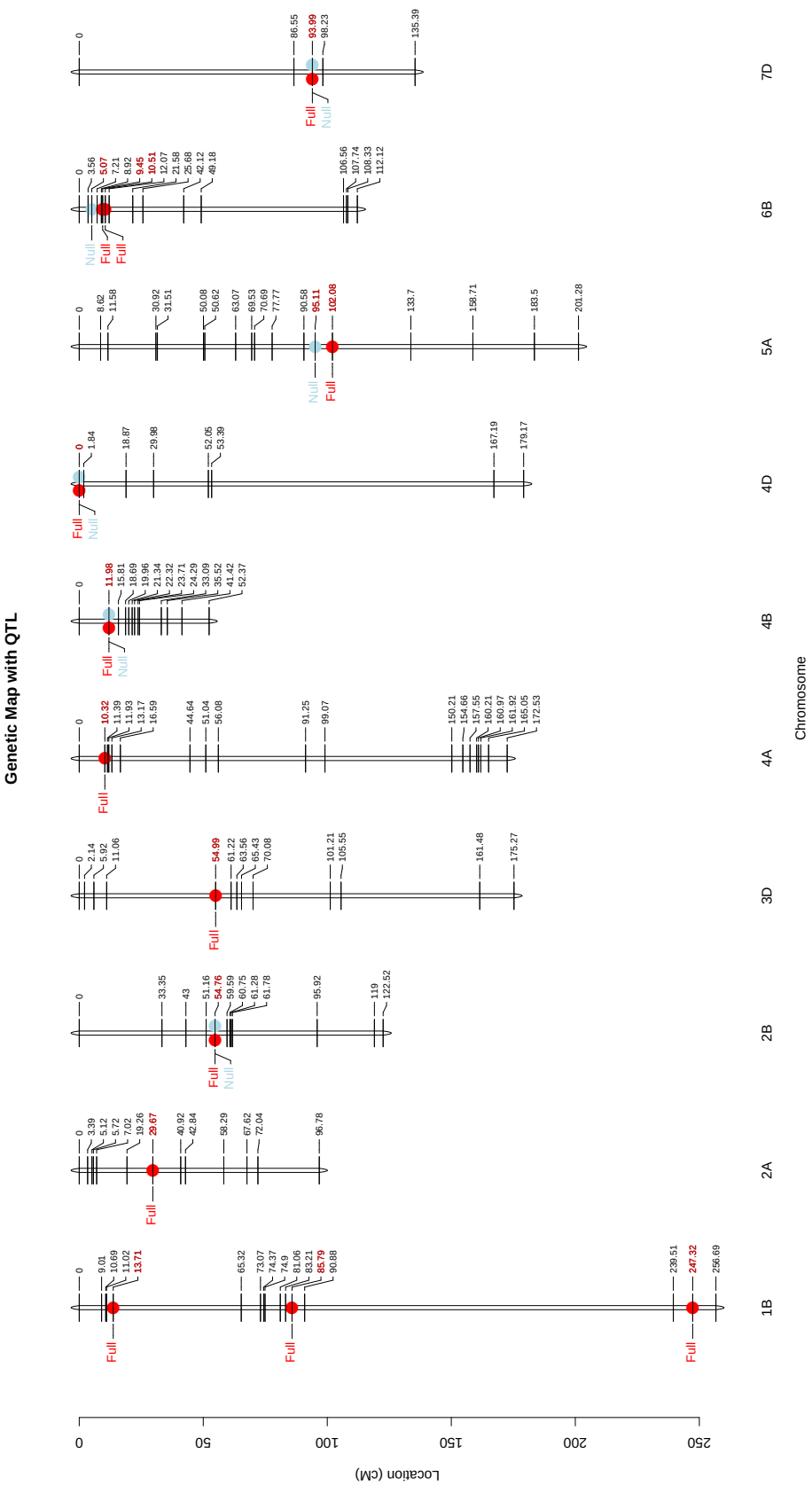

Figure 4.10: Genetic map with QTL for the Full and Null models. Marker QTL are highlighted according to user specifications.

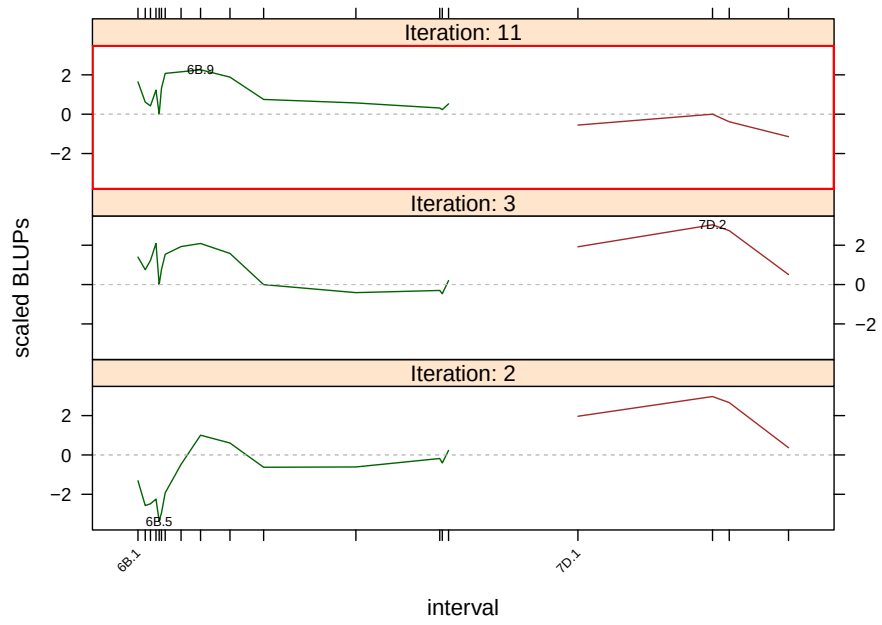

Figure 4.11: Scaled BLUPs of the interval QTL effects for iterations 2, 3 and 11 for the QTL analysis involving the full model.

between the QTL. Consequently, the dubiousness in selecting the second QTL increases as the QTL are more tightly linked. This can be alleviated by choosing an exclusion window around the region of the first selected QTL. In the next analysis an exclusion window of 20cM is added and the `method = "random"` formulation will be used

```
R> st.qtlFR <- wgaim(st.fmF, phenoData = phenoSxT, intervalObj = genoSxT,
+     merge.by = "id", gen.type = "interval", method = "random",
+     selection = "interval", trace = "fullmodel.txt", exclusion.window = 20)
```

Its summary is then

```
R> summary(st.qtlFR, genoSxT, LOD = FALSE)
```

```
  Chromosome Left Marker dist(cM) Right Marker dist(cM)   Size  Prob % Var
1         1B      Glu.B3    11.02  P34.M51.286    13.71  0.159 0.003   1.5
2         1B       gwm11    90.88       gwm140   239.51 -0.743 0.000   5.8
3         2A       wmc198   29.67       wmc170    40.92 -0.182 0.003   1.8
4         2B    wmc474USQ   54.76       wmc35a    59.59  0.809 0.000  34.5
5         3D    TeloPAGG2   54.99    TeloPAGG1    61.22 -0.158 0.005   1.4
6         4A       germin   10.32       cdo795    11.39  0.138 0.008   1.2
7         4B      barc193       0        csME1    11.98 -0.436 0.000   9.4
8         4D     Rht2.mut       0        csME2     1.84  0.285 0.000   4.6
```

```
9          5A   PAACTelo2    95.11    P46.M37.4   102.08 -0.323 0.000   5.4
10         6B      cdo507     8.92      barc354     9.45 -0.444 0.000  11.0
11         7D      gwm437    86.55         wmc94    93.99  0.285 0.000   4.3
```

The random QTL interval analysis summary output is identical to the fixed QTL interval summary output with the exception of the significance of the QTL. From section 2.4.2 the significance or strength of QTL are now determined by a probability statement. Values displayed here then measure the probability that a QTL is actually zero and allow an interpretation similar to a p-value. In comparison to the interval QTL analysis, this summary indicates only one QTL was selected on chromosome 6B and three less QTL were found in total. However, all QTL found from the interval random effects analysis were shared with the interval fixed effects analysis.

Bibliography

Ball, R. (2010). ***BayesQTLBIC**: Bayesian Multi-Locus QTL Analysis Based on the BIC Criterion*. R package version 1.0-1.

Beavis, W. D. (1994). The power and deceit of QTL experiments: lessons from comparative QTL studies. In *Proceedings of the Forty-Ninth Annual Corn and Sorghum Industry Research Conference*, pages 250–266. American Seed Trade Association, Washington, DC.

Beavis, W. D. (1998). QTL analyses: power, precision and accuracy. In Patterson, A. H., editor, *Molecular Dissection of Complex Traits*, pages 145–162. CRC Press, New York.

Bennett, D., Izanloo, A., Edwards, J., Kuchel, H., Chalmers, K., Tester, M., Reynolds, M., Schnurbusch, T., & Langridge, P. (2012a). Identification of novel quantitative trait loci for days to ear emergence and flag leaf glaucousness in a bread wheat (triticum aestivum l.) population adapted to southern australian conditions. *Theoretical and Applied Genetics* **124**, 697–711.

Bennett, D., Izanloo, A., Reynolds, M., Kuchel, H., Langridge, P., & Schnurbusch, T. (2012b). Genetic dissection of grain yield and physical grain quality in bread wheat (triticum aestivum l.) under water-limited environments. *Theoretical and Applied Genetics* **125**, 255–271.

Bennett, D., Reynolds, M., Mullan, D., Izanloo, A., Kuchel, H., Langridge, P., & Schnurbusch, T. (2012c). Detection of two major grain yield qtl in bread wheat (triticum aestivum l.) under heat, drought and high yield potential environments. *Theoretical and Applied Genetics* **125**, 1473–1485.

Bonneau, J., Taylor, J. D., Parent, B., Reynolds, M., Feuillet, C., Langridge, P., & Mather, D. (2012). Mult-environment analysis and fine mapping of a yield related QTL on chromosome 3B of wheat. *Theoretical and Applied Genetics, Accepted* **126**, 747–761.

Broman, K. W. & Sen, S. (2009). *A Guide to QTL Mapping with R/**qtl***. Springer-Verlag. ISBN: 978-0-387-92124-2.

BIBLIOGRAPHY

Broman, K. W. & Wu, H. (2014). ***qtl**: Tools for Analayzing QTL Experiments*. R package version 1.33-7.

Gilmour, A. R. (2007). Mixed Model Regression Mapping for QTL Detection in Experimental Crosses. *Computational Statistics and Data Analysis* **51**, 3749–3764.

Gilmour, A. R., Gogel, B. J., Cullis, B. R., & Thompson, R. (2009). ***ASReml** User Guide*. Release 3.0.

Gilmour, A. R., Thompson, R., & Cullis, B. R. (1995). Average information REML: An efficient algorithm for variance parameter estimation in linear mixed models. *Biometrics* **51**, 1440–1450.

Gogel, B. J. (1997). *Spatial analysis of multi-environment variety trials*. PhD thesis, Department of Statistics, University of Adelaide.

Gogel, B. J., Welham, S. J., Verbyla, A. P., & Cullis, B. R. (2001). Outlier detection in linear mixed effects; summary of research. report p106. Technical report, University of Adelaide, Biometrics.

Hayley, C. S. & Knott, S. A. (1992). A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. *Heredity* **69**, 315–324.

Huang, B. & George, A. (2009). Look before you leap: a new approach to mapping qtl. *TAG Theoretical and Applied Genetics* **119**, 899–911. 10.1007/s00122-009-1098-y.

Kang, H. M., Zaitlen, N. A., Wade, C. M., Kirby, A., Heckerman, D., Daly, M. J., & Eskin, E. (2008). Efficient control of population structure in model organism association mapping. *Genetics* **178**, 1709–1723.

Lander, E. & Green, P. (1987). Construction of multilocus genetic linkage maps in humans. *Proceedings of the National Academy of Science* **84**, 2363–2367.

Martinez, O. & Curnow, R. N. (1992). Estimating the locations and sizes of the effects of quantitative trait loci using flanking markers. *Theoretical and Applied Genetics* **85**, 480–488.

Melchinger, A. E., Utz, H. F., & Schon, C. C. (1998). Quantitative trait loci (QTL) mapping using different testers and independent population samples in maize reveals low power of QTL detection and large bias in estimates of QTL effects. *Genetics* **149**, 383–403.

Oakey, H., Verbyla, A. P., S, P. W., Cullis, B. R., & Kuchel, H. (2006). Joint Modelling of Additive and Non-Additive Genetic Line Effects in Single Field Trials. *Theoretical and Applied Genetics* **113**, 809–819.

Patterson, H. D. & Thompson, R. (1971). Recovery of interblock information when block sizes are unequal. *Biometrika* **58**, 545–554.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics* **6**, 461–464.

Seaton, G., Haley, C. S., Knott, S. A., Kearsey, M., & Visscher, P. M. (2002). QTL Express: mapping quantitative trait loci in simple and complex pedigrees. *Bioinformatics* **18**, 339–340.

Shriner, D. & Yi, N. (2009). Deviance Information Criterion (DIC) in Bayesian Multiple QTL Mapping. *Computational Statistics and Data Analysis* **53**, 1850–1860.

Smith, A., Cullis, B. R., & Thompson, R. (2001). Analysing variety by environment data using multiplicative mixed models. *Biometrics* **57**, 1138–1147.

Smith, A., Cullis, B. R., & Thompson, R. (2005). The analysis of crop cultivar breeding and valuation trials: An overview of current mixed model approaches. *Journal of Agricultural Science* **143**, 449–462.

Smith, A. B., Lim, P., & Cullis, B. R. (2006). The design and analysis of multi-phase plant breeding programs. *Journal of Agricultural Science* **144**, 393–409.

Stram, D. O. & Lee, J. W. (1994). Variance components testing in the longitudinal mixed effects model. *Biometrics* **50**, 1171–1177.

Verbyla, A. P. (1990). A conditional derivation of residual maximum likelihood. *Australian Journal of Statistics* **32**. 227-230.

Verbyla, A. P., Cullis, B. . R., & Thompson, R. (2007). The analysis of QTL by simultaneous use of the of the full linkage map. *Theoretical and Applied Genetics* **116**, 95–111.

Verbyla, A. P., Taylor, J. D., & Verbyla, K. L. (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research* **94**, 291–306.

Whittaker, J. C., Thompson, R., & Visscher, P. M. (1996). On the mapping of QTL by regression of phenotype on marker-type. *Heredity* **77**, 22–32.

Xu, S. (2003). Estimating polygenic effects using markers of the entire genome. *Genetics* **164**, 789–801.

Yandell, B. S., Mehta, T., Banerjee, S., Shriner, D., Venkataraman, R., Moon, J. Y., Neely, W. W., Wu, H., Smith, R., & Yi, N. (2005). R/**qtlbim**: QTL with Bayesian Interval Mapping in Experimental Crosses. *Bioinformatics* **23**, 641–643.

BIBLIOGRAPHY

Zeng, Z.-B. (1994). Precision mapping of quantitative trait loci. *Genetics* **136**, 1457–1468.

Zhang, M., Zhang, D., & Wells, M. (2008). Variable selection for large p small n regression models with incomplete data: mapping QTL with epistases. *BMC Bioinformatics* **9**.

Zhou, Q. (2010). A Guide to QTL Mapping with R/**qtl**. *Journal of Statistical Software, Book Reviews* **32**, 1–3.