NIASRA

NATIONAL INSTITUTE FOR APPLIED STATISTICS RESEARCH AUSTRALIA

National Institute for Applied Statistics Research Australia

University of Wollongong, Australia

Working Paper

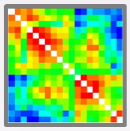
17-11

Extensions in Linear Mixed Models and Design of Experiments

David Butler, Brian Cullis, and Julian Taylor

National Institute for Applied Statistics Research Australia, University of Wollongong,
Wollongong NSW 2522, Australia Phone +61 2 4221 5435, Fax +61 2 4221 4998.
Email: karink@uow.edu.au

GRDC
Grains Research & Development Corporation



SAGI

Statistics for the Australian Grains Industry

Extensions in Linear Mixed Models and Design of Experiments

David Butler[1] & Brian Cullis[2,3] & Julian Taylor[4]

[1]Principal Biometrician, Plant Science, Agri-Science QLD, DEEDI

[2]Professor of Biometry, School of Mathematics & Applied Statistics, Faculty of Informatics, University of Wollongong

[3]CMIS, Environmental Informatics & Food Futures Flagship

[4]Research Scientist, School of Food Agriculture & Wine, University of Adelaide

Workshop in conjunction with:

Biometrics by the Blowholes

Presenter Affiliations:

CSSM Centre for Statistical & Survey Methodology University of Wollongong

CSIRO

Queensland Government

THE UNIVERSITY of ADELAIDE

Contents

Chapter

B The design and analysis of variety trials using mixtures of composite and individual plot samples.

1 Background

Field trials for evaluating plant varieties are conducted in order to obtain information on a range of traits including grain yield and grain quality traits, for example as in the National Variety Trials (NVT) system used for obtaining information on late stage evaluation of cereal varieties in Australia. Some grain quality traits are costly to measure so that it is not possible to test all plots individually. It has therefore become common practice to test a single composite sample for each variety in the trial but this precludes the use of an efficient statistical analysis. Smith et al. (2011) proposed an approach in which a proportion of varieties be tested using individual replicate samples. The remaining varieties may either be tested as composite samples or with reduced replication. This allows application of efficient mixed model analyses for both the inexpensive traits (using data from the complete set of individual replicate plots) and expensive traits (using data from a subset of the plots or a mixture of composite and individual plot samples). In the first chapter of these notes we present the key results of Smith et al. (2011) beginning with a brief review of statistical analysis methods for the analysis of single and multi-environment field trials. We then consider two extensions namely the notion of an embedded field design and secondly an analysis of composite and individual plot samples. This is initially done in a single phase experiment. We consider the extension of this idea to multi-phase experiments in 3

1.1 Statistical analysis methods for field trials

The literature on methods for the analysis of field trials (which includes the specific setting of variety trials) is quite expansive but the methods can be broadly classified as either randomisation or model based. In the former the model for the random and residual effects is determined purely from the block structure whereas in the latter it is either assumed or

selected with the objective of providing a good fit to the data. Model based approaches for the analysis of variety trials aim to account for the effect of spatial heterogeneity on the prediction of genotype contrasts. Typically the heterogeneity reflects the fact that, in the absence of design effects (that is, treatment and block effects), data from plots that are close together (that is, neighbouring plots) are more similar (positively correlated) than those that are further apart. Numerous authors have proposed analytical methods to remove the effects of such trend. We use the approach originally proposed by Gilmour et al. (1997) and extended by Stefanova et al. (2009). These authors assume that field trials are arranged as rectangular arrays indexed by rows and columns (extensions to other arrangements are straight-forward). Gilmour et al. (1997) extended the approach of Cullis & Gleeson (1991) by partitioning spatial variation into two types of smooth trend (local and global) and extraneous variation. Local trend reflects, for example, small scale soil depth and fertility fluctuations. Global trend reflects non-stationary trend across the field. Extraneous variation is often linked to trial management, in particular, procedures that are aligned with the field rows and columns (for example, the sowing and harvesting of plots). Certain procedures may result in row and column effects (systematic and/or random). In the Gilmour et al. (1997) approach global trend and extraneous variation are accommodated in the mixed model by including appropriate fixed and/or random effects. Local stationary trend is modelled using a covariance structure for the residuals. A plausible model that has broad application for two-dimensional (row by column) field trials is a separable autoregressive process of order 1 (here-after denoted AR1×AR1) as originally proposed by Cullis & Gleeson (1991) and used by Gilmour et al. (1997) and Stefanova et al. (2009).

For complete generality we consider a series of t trials (synonymous with environments) in which a total of m varieties has been grown (without necessarily all varieties being grown in all trials). First we examine the statistical model used for the analysis of a single variety trial then build on this to describe the analysis for the complete series of trials.

1.1.1 Single trial analysis

We consider the analysis of the j^{th} trial ($j = 1 \dots t$) within the series and assume it comprises n_j plots laid out in a rectangular array of c_j columns by r_j rows (so that $n_j = c_j r_j$). Let $\boldsymbol{y}_j$ be the $n_j \times 1$ vector of data, ordered as rows within columns. The statistical model for $\boldsymbol{y}_j$ can be written as

$$\boldsymbol{y}_j = \boldsymbol{X}_j \boldsymbol{\tau}_j + \boldsymbol{Z}_{\boldsymbol{g}_j} \boldsymbol{u}_{\boldsymbol{g}_j} + \boldsymbol{Z}_{\boldsymbol{p}_j} \boldsymbol{u}_{\boldsymbol{p}_j} + \boldsymbol{e}_j \tag{1}$$

where $\boldsymbol{\tau}_j$ is a vector of fixed effects with associated design matrix $\boldsymbol{X}_j$ (assumed to have full column rank); $\boldsymbol{u}_{\boldsymbol{g}_j}$ is the $m \times 1$ vector of random variety effects with associated design matrix $\boldsymbol{Z}_{\boldsymbol{g}_j}$; $\boldsymbol{u}_{\boldsymbol{p}_j}$ is a vector of random non-genetic (or peripheral) effects with associated design matrix $\boldsymbol{Z}_{\boldsymbol{p}_j}$ and $\boldsymbol{e}_j$ is the vector of residuals for the trial. In the simplest case the vector $\boldsymbol{\tau}_j$ comprises an overall mean (intercept) for the trial but may include effects to

Table 1: Trial layout information for canola data: numbers of rows, columns and entries and proportion p of entries with two replicates. Trial mean oil content also presented.

Trial	Rows	Columns	Entries	Mean oil	p
1	48	6	213	38.2	0.352
2	51	6	232	43.9	0.319
3	52	6	245	40.6	0.273
4	52	6	252	46.0	0.238
5	53	6	254	45.6	0.252
6	49	6	220	38.9	0.336
7	53	6	260	47.5	0.223

accommodate spatial trend (see below). The effects $\boldsymbol{u}_{\boldsymbol{p}_j}$ include effects for major blocking factors associated with the trial design and possibly effects to accommodate spatial trend (see below). For simplicity we assume independent variety effects with $var\left(\boldsymbol{u}_{\boldsymbol{g}_j}\right) = \sigma^2_{g_j}\boldsymbol{I}_m$ where $\sigma^2_{g_j}$ is the genetic variance for trial j. Other genetic variance models are possible, in particular a known relationship matrix may be incorporated (see Oakey et al., 2007). Note that if fewer than m varieties were grown in the j^{th} trial then $\boldsymbol{Z}_{\boldsymbol{g}_j}$ will contain some zero columns.

In terms of the residuals we follow the spatial modelling approach of Stefanova et al. (2009) in which a separable autoregressive process of order one (denoted AR1×AR1) is assumed so that $var\left(\boldsymbol{e}_j\right) = \boldsymbol{R}_j = \sigma^2_j\boldsymbol{\Sigma}_{\boldsymbol{c}j} \otimes \boldsymbol{\Sigma}_{\boldsymbol{r}j}$ where $\boldsymbol{\Sigma}_{\boldsymbol{c}j}$ and $\boldsymbol{\Sigma}_{\boldsymbol{r}j}$ are the $c_j \times c_j$ and $r_j \times r_j$ correlation matrices for the column and row dimensions (so each is a function of a single autocorrelation parameter, that is, ρ_{c_j} and ρ_{r_j} respectively).

The spatial analysis process is sequential, commencing with the fitting of a base-line model followed by the application of diagnostics to assess model adequacy. In particular the presence of outliers, non-stationary global trend and extraneous variation is investigated (see Stefanova et al., 2009). If there is evidence of the latter it may be accommodated in the model by including appropriate effects in either $\boldsymbol{\tau}_j$ or $\boldsymbol{u}_{\boldsymbol{p}_j}$.

1.1.2 MET analysis

We now consider the full series of t trials and let $\boldsymbol{y} = (\boldsymbol{y}'_1, \boldsymbol{y}'_2, \ldots \boldsymbol{y}'_t)'$ denote the vector of individual plot data combined across trials. The MET model proposed by Smith et al. (2001) for $\boldsymbol{y}$ can be written as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_{\boldsymbol{g}}\boldsymbol{u}_{\boldsymbol{g}} + \boldsymbol{Z}_{\boldsymbol{p}}\boldsymbol{u}_{\boldsymbol{p}} + \boldsymbol{e} \tag{2}$$

where $\boldsymbol{\tau} = (\boldsymbol{\tau}'_1, \boldsymbol{\tau}'_2, \ldots \boldsymbol{\tau}'_t)'$ is a vector of fixed effects with associated design matrix $\boldsymbol{X} = diag\left(\boldsymbol{X}_j\right)$; $\boldsymbol{u}_{\boldsymbol{p}} = (\boldsymbol{u}'_{\boldsymbol{p}1}, \boldsymbol{u}'_{\boldsymbol{p}2}, \ldots \boldsymbol{u}'_{\boldsymbol{p}t})'$ with associated design matrix $\boldsymbol{Z}_{\boldsymbol{p}} = diag\left(\boldsymbol{Z}_{\boldsymbol{p}_j}\right)$ and variance matrix $var\left(\boldsymbol{u}_{\boldsymbol{p}}\right) = \boldsymbol{G}_{\boldsymbol{p}}$; $\boldsymbol{e} = (\boldsymbol{e}'_1, \boldsymbol{e}'_2, \ldots \boldsymbol{e}'_t)'$ with associated variance matrix $var\left(\boldsymbol{e}\right) = \boldsymbol{R} = diag\left(\boldsymbol{R}_j\right)$.

The vector $\boldsymbol{u}_{\boldsymbol{g}} = (\boldsymbol{u}'_{\boldsymbol{g}1}, \boldsymbol{u}'_{\boldsymbol{g}2}, \ldots \boldsymbol{u}'_{\boldsymbol{g}t})'$ is the $mt \times 1$ vector of variety effects for individual trials with associated design matrix $\boldsymbol{Z}_{\boldsymbol{g}} = diag\left(\boldsymbol{Z}_{\boldsymbol{g}_j}\right)$. We assume that $var\left(\boldsymbol{u}_{\boldsymbol{g}}\right) =$

Table 2: Residual log-likelihoods for MET models fitted to canola data

Model for $\boldsymbol{G_e}$	Variance parameters genetic	total	Residual log-likelihood
1. Diagonal	7	37	-1679.89
2. Uniform	2	32	-1179.61
3. FA1	14	46	-1115.38
4. FA2	20	52	-1103.58

$\boldsymbol{G_e} \otimes \boldsymbol{I}_m$ where $\boldsymbol{G_e}$ is the $t \times t$ genetic variance matrix across trials (that is, with diagonal elements given by the genetic variances for each trial and the off-diagonal elements the genetic covariances between pairs of trials). Here we have assumed independence between genotypes but as noted for single trial analysis the use of a relationship matrix is also possible. Various forms for $\boldsymbol{G_e}$ may be used. We follow the approach of Smith et al. (2001) who advocate the use of factor analytic (FA) models. The FA model for $\boldsymbol{u_g}$ is given by:

$$\boldsymbol{u_g} = (\boldsymbol{\Lambda} \otimes \boldsymbol{I}_m)\boldsymbol{f} + \boldsymbol{\delta} \tag{3}$$

where $\boldsymbol{\Lambda}$ is a $t \times k$ matrix of trial loadings (k being the number of factors included in the model), $\boldsymbol{f}$ is an $mk \times 1$ vector of variety scores and $\boldsymbol{\delta}$ is the $mt \times 1$ vector of residual genetic effects. We assume $var(\boldsymbol{f}) = \boldsymbol{I}_{mk}$ and $var(\boldsymbol{\delta}) = \boldsymbol{\Psi} \otimes \boldsymbol{I}_m$ where $\boldsymbol{\Psi} = diag(\psi_j)$ where ψ_j is known as the specific variance for the j^{th} trial. It then follows that

$$\boldsymbol{G_e} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{\Psi}$$

1.2 Analysis of example

Here we consider oil content data (measured using NIR as percentage of whole grain) from a series of 7 canola breeding trials (data kindly supplied by Canola Breeders Western Australia Pty Ltd). The number of entries (synonymous with varieties) grown in individual trials ranged from 213 to 260 (Table 1) with a total of 260 entries across all trials. Entries were mainly breeding lines but some commercial varieties were also included. Each trial was laid out as a rectangular array of plots indexed by rows and columns (Table 1). The design for each trial was a partially replicated ($p-$rep) design (Cullis et al., 2006). In these designs a proportion p of entries is replicated with the remainder having only single plots. The value of p for these trials ranged from 0.223 to 0.352 (Table 1). Each design was resolvable in the sense that all replicated entries occurred once in the first block (columns 1-3) and then again in the second block (columns 4-6).

In the Smith et al. (2001) approach for analysis the first step is to determine appropriate spatial models for individual trials. This is most efficiently achieved from a computational perspective by fitting a simple form for the genetic variance matrix in equation (2). We often use a diagonal form, that is, $\boldsymbol{G_e} = diag\left(\sigma^2_{g_j}\right)$ since this is analogous to analysing the data for each trial separately. The spatial models so determined can then be re-checked once the final genetic model has been fitted. For the canola data

Table 3: Estimates of genetic variance parameters from FA2 model fitted to canola data.

	FA parameters			Genetic covariance/correlation matrix						
Trial	$\boldsymbol{\lambda}_1$	$\boldsymbol{\lambda}_2$	$\boldsymbol{\Psi}$	1	2	3	4	5	6	7
1	0.762	0.258	0.296	0.944	0.76	0.71	0.71	0.77	0.79	0.69
2	1.163	0*	0.076	0.886	1.429	0.89	0.89	0.89	0.85	0.93
3	0.986	−0.003	0.201	0.750	1.146	1.173	0.84	0.84	0.79	0.87
4	1.463	−0.040	0.395	1.104	1.701	1.442	2.536	0.84	0.79	0.88
5	1.259	0.278	0.214	1.031	1.464	1.240	1.830	1.877	0.88	0.83
6	0.868	0.410	0.072	0.767	1.009	0.854	1.253	1.207	0.993	0.75
7	1.284	−0.272	0.100	0.908	1.493	1.266	1.888	1.541	1.003	1.822

* Parameter not estimated but fixed at this value to ensure a unique solution

the base-line model includes random block effects for each trial (included in the vector $\boldsymbol{u_p}$) with a separate block variance for each trial and a separate spatial covariance model for the errors for each trial. Following the fit of this base-line model the use of diagnostics as described in Stefanova et al. (2009) revealed the need to add linear regressions on row number for the first 3 trials (added to the model as fixed effects in $\boldsymbol{\tau}$) and random column effects for trials 2 and 6 (added to the model as extra random effects in $\boldsymbol{u_p}$).

Having determined acceptable spatial models we investigate more appropriate models for the variety effects across trials. The diagonal model for $\boldsymbol{G_e}$ allows for a separate genetic variance for each trial but assumes the variety effects in different trials to be uncorrelated. The simplest correlation model that is often applied (implicitly) to MET data is the uniform model that assumes a common genetic correlation for all pairs of trials and a common genetic variance. The residual log-likelihoods from fitting the uniform and diagonal models are given in Table 2.

We then fitted factor analytic models for the genetic variance structure. The residual log-likelihood from fitting an FA model with one factor (denoted FA1) to the canola data is given in Table 2. Since the uniform model is nested within the FA1 model they can be formally compared using a residual maximum likelihood ratio test (REMLRT). The FA1 model provided a significantly better ($p < 0.001$) fit to the data. An FA2 model was then fitted and a REMLRT revealed this to be significantly better ($p < 0.001$) than the FA1 model. Note that in order to ensure uniqueness of the loading matrix in the FA2 model it is necessary to impose a single constraint (Smith et al., 2001). We chose to constrain the second element in the second loading to be equal to zero. This is purely for computational reasons and has no biological basis. If a meaningful interpretation of the loadings is required we usually rotate the solution. With only 7 trials it is not possible to fit higher order FA models so we conclude that the FA2 model provides the best fit to these data.

Estimates of the variance parameters and fixed effects from the FA2 model are given in Tables 3 and 4. The estimated genetic variances range from 0.94 to 2.54 and the genetic correlations are very strong, ranging from 0.69 to 0.93 (Table 3). Examination of

Table 4: Estimates of non-genetic fixed effects and variance parameters from FA2 model fitted to canola data

	Fixed effects			Variances		Autocorrelations	
Trial	Intercept	lin(row)	block	column	residual	column	row
1	38.2	0.012	0.104		0.326	0.13	0.39
2	43.9	−0.044	0.087	0.306	0.377	0.20	0.45
3	40.6	−0.018	0.082		0.594	0.14	0.59
4	45.9		0.000		1.282	0.35	0.78
5	45.7		0.271		2.217	0.26	0.61
6	38.9		0.000	0.123	0.478	0.27	0.55
7	47.6		0.000		0.707	0.21	0.56

Table 4 shows the existence of substantial non-genetic variation in most of the trials. As already discussed there was significant linear trend across rows for trials 1-3 and significant variation between columns for trials 2 and 6. Variation between blocks was large for trial 5. Stationary spatial trend was evident in most trials with the largest autocorrelation parameters associated with trial 4.

The final step in the analysis was to obtain E-BLUPs of the variety effects for each trial. Of particular relevance to this paper is the impact of using the Smith et al. (2001) approach to analysis on these predictions. Figure 1 contains pairwise plots of the predicted variety effects for trial 5 obtained using three methods, namely (a) variety averages of raw data (expressed as deviations from the trial mean); (b) E-BLUPs, $\tilde{\boldsymbol{u}}_{g_5}$, from model 1 in Table 2 (diagonal form for $\boldsymbol{G_e}$); (c) E-BLUPs, $\tilde{\boldsymbol{u}}_{g_5}$, from model 4 in Table 2 (FA2 form for $\boldsymbol{G_e}$). There are large changes in moving from (a) to (b) that are the result of having modelled spatial variation. Note that method (a) is analogous to the practice of measuring composite samples for each entry. Then there are also substantial changes in moving from (b) to (c) that are the result of incorporating information from other (highly correlated) trials.

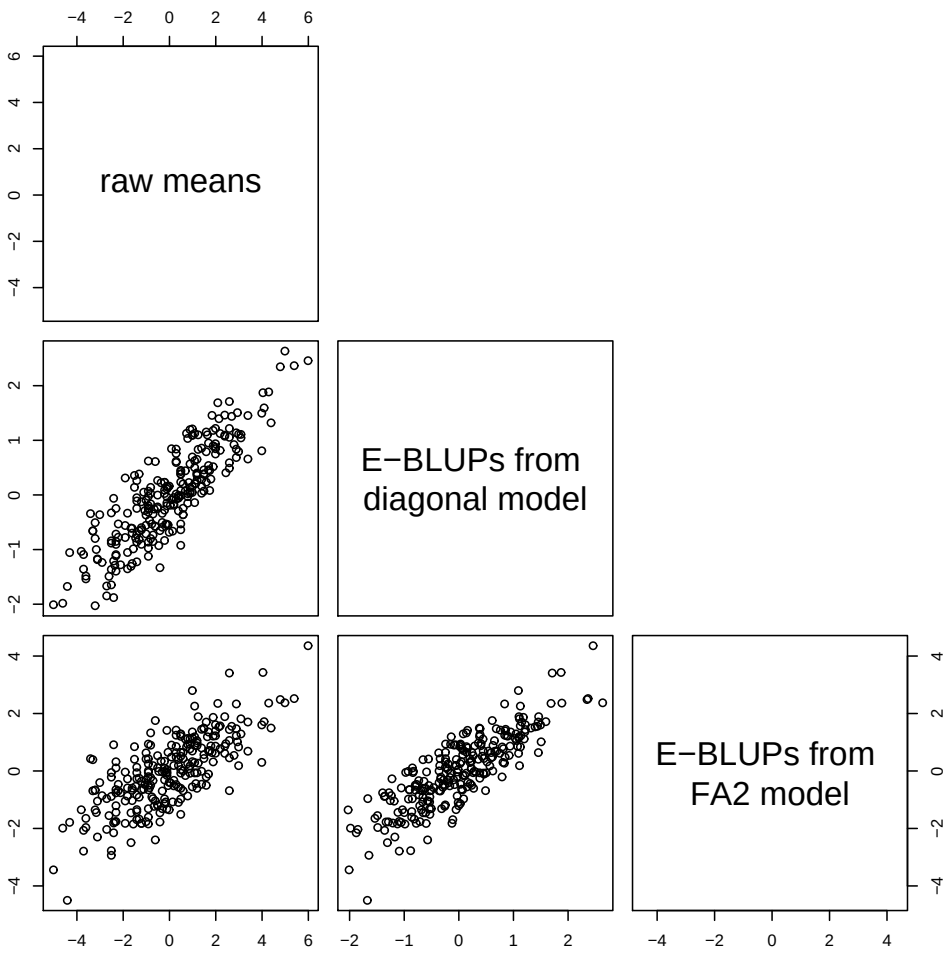


Figure 1: Predicted variety effects for trial 5 obtained using three methods.

2 Reducing replication for expensive traits

There may be traits for which it is prohibitively expensive to take measurements on all individual replicate plots in a trial. We assume that we are limited to testing s_j samples. Currently it is common practice (for example in NVT) to use $s_j = m$ samples where each sample is a composite of all b_j replicate plots for a variety. We propose schemes that lie between the two extremes of using all composite samples ($s_j = m$) and all individual plot samples ($s_j = n_j$). In order to describe the schemes we consider one of the examples presented by Smith et al. (2011).

This example is similar to that of the real breeding program example of Section 1.2 in which we wish to test 90 canola entries for grain yield and oil content. Unlike the real example, however, we now assume a fully replicated design (with two replicates) for measuring grain yield making a total of 180 plots. We assume that these are laid out in a similar manner to the real example with 6 columns and 30 rows and with the first block comprising columns 1-3 and the second block columns 4-6. In order to test oil content we assume we are limited to testing 120 samples.

2.1 Two replicate example

In order to reduce the number of samples from $s_j = 180$ to $s_j = 120$ we propose to use replicate plots of 30 entries only. The remaining 60 entries may either be tested as single plots or as composite samples formed from the two replicate plots of these entries. These approaches are discussed in turn in the following sections.

2.1.1 Using subsets of plots

In this scenario the remaining 60 entries are tested as single plots so that a total of 120 samples is tested and each sample corresponds to a different plot. In terms of the subset of plots to be tested one approach, here-after termed "subset selection", would be to generate an efficient design for grain yield then select a subset of plots for testing oil content without regard to their spatial location. We will call the overall layout an OT1 design to reflect the fact that it has been optimised with respect to a single trait (usually grain yield) only. It is usually desirable for field designs to comprise a contiguous set of plots so that subset selection is not ideal for oil content. As an example we may proceed as in Figure 2 in which all plots in the first block will be quality tested and then 30 plots in the second block (corresponding to the pre-chosen set of replicated entries for this trial).

An alternative approach is to restrict the subset of plots for oil content to occupy a rectangular sub-section of the overall layout (and thence a contiguous set of plots). In this example in which the full design comprises two replicates we note that the subsetting of plots creates a replication pattern synonymous with a $p-$rep design. We may therefore a priori create an embedded design that is a valid $p-$rep design in its own right. In our example we chose to locate the $p-$rep design within the sub-section corresponding

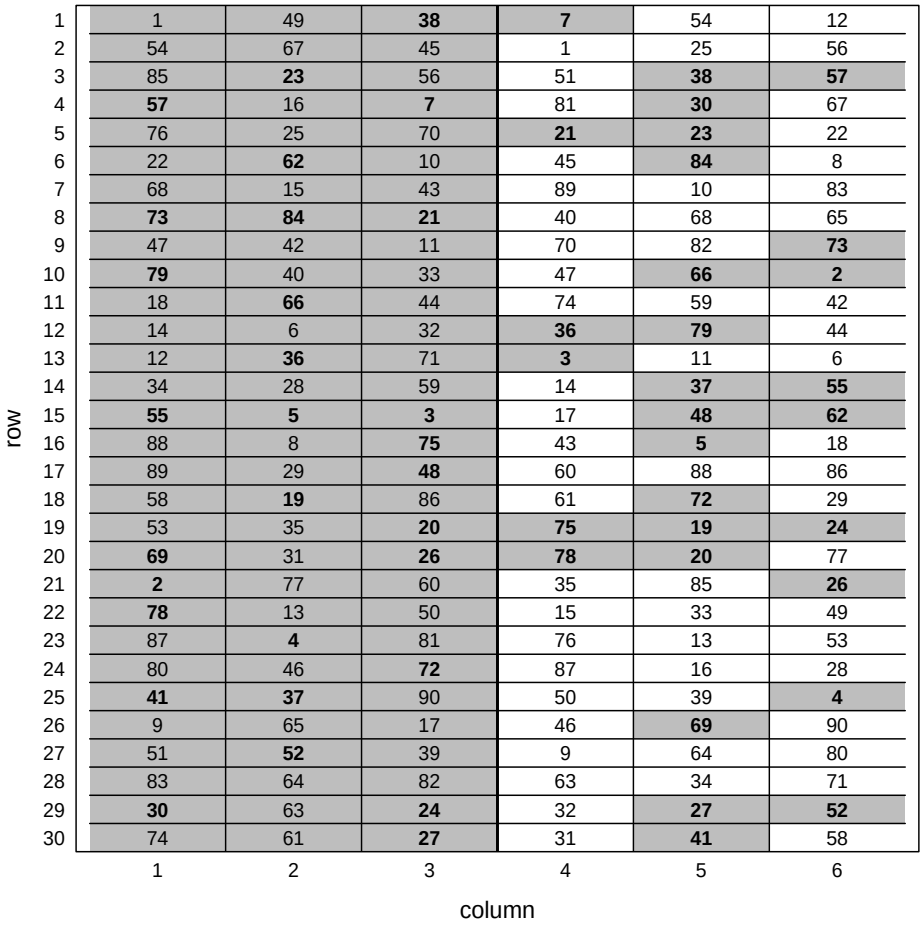


Figure 2: OT1 design for testing grain yield and oil content. Full layout for grain yield arranged as 30 rows by 6 columns with 2 replicates of 90 entries (block 1 = columns 1-3; block 2 = columns 4-6). Entry numbers are given for each plot. The subset of plots for testing oil content (shaded plots) comprises 2 replicates of 30 entries (numbers in bold face) and single replicates of 60 entries (numbers in plain face).

to the first 20 rows and all 6 columns. This will be called the "embedded area" and the remaining plots comprise the "non-embedded area".

In this way we may create a design that is as efficient as possible for both grain yield and oil content. This can be achieved in a number of ways, two of which include:

1. an efficient $p-$rep design is generated for the embedded area, then conditional on this a design is generated for the non-embedded area, or

2. the designs for the embedded and non-embedded areas are generated simultaneously

where in each case the varieties to be replicated in the embedded area are predetermined. The designs for the simulation study in Section 2.2 were generated under the second approach using software currently under development for the R statistical environment (R Development Core Team, 2011), and detailed below.

It is widely accepted that data from individual field trials often exhibits spatial correlation so may be analysed using models of the form in equation (1). A logical step has been to accommodate spatial correlation in the design phase (see Martin & Eccleston, 1997; Eccleston & Chan, 1998, for example). We adopt this approach here.

Given a (prespecified) replication scheme, potential extraneous effects and a residual correlation structure, an interchange algorithm governed the assignment of varieties to plots in order to optimise a specific design criterion. For this study, the variety effects were considered fixed and the criterion chosen was *A-optimality*, where the average pairwise variance of variety effects is minimised. Computationally, the *A-value* is calculated using the mixed model equations corresponding to the pre-specified form of mixed model relevant to the design problem. In our example the mixed model comprised fixed effects for the overall mean and varieties and random effects for blocks, rows, columns and a factor with two levels indicating whether the plot is in the embedded or non-embedded area. A separable AR1×AR1 correlation structure was assumed for the residuals. The autocorrelations were set at 0.3 and 0.6 for the column and row dimensions respectively, the variance ratios for all random effects were set to 0.1 and without loss of generality the residual variance was fixed at 1.0. These parameter values reflect our experience in analysing grain yield data from thousands of trials across a number of crop types. Although the corresponding values for grain quality traits are as yet largely unknown we note that the designs are fairly robust to mis-specification of the parameters (Eccleston & Chan, 1998) so that the values given here provide a reasonable starting point. The *A-value* is computed from the coefficients matrix for varieties, adjusted for all other effects by applying the matrix sweep operator (Gentle, 1998) to the mixed model equations.

The design is resolvable for replicated varieties, and the permutation of varieties to plots was restricted to be within each of the embedded and non-embedded areas, while respecting this resolvability. The design so constructed ensures that all 30 replicated varieties in the first 20 rows occurred once in the first block (columns 1-3) and again in the second block (columns 4-6). Figure 3 shows one realisation of this process. We will call the overall layout an OT2 design to reflect the fact that it has been optimised with respect to two traits. Note that the replicated varieties for testing oil content in Figure 2 are identical to those in Figure 3.

In the context of METs, it is important to consider design for the complete series rather than individual trial designs in isolation. This is particularly so for $p-$rep designs (including embedded $p-$rep designs) in the context of developing a (partial) replication scheme. To illustrate, we extend our scenario once again to match the real example in terms of a series of 7 trials. In this setting we wish to construct the embedded $p-$rep designs such that entries are as equally replicated across trials as possible. For our scenario this results in a series of $p-$rep designs in which entries have a total of either 9 (60 entries) or 10 (30 entries) replicates across all trials. We can achieve a (partially) balanced allocation of replication levels by regarding the MET as an incomplete block design with 7 replicates (corresponding to trials) of 90 treatments, and two blocks per replicate of

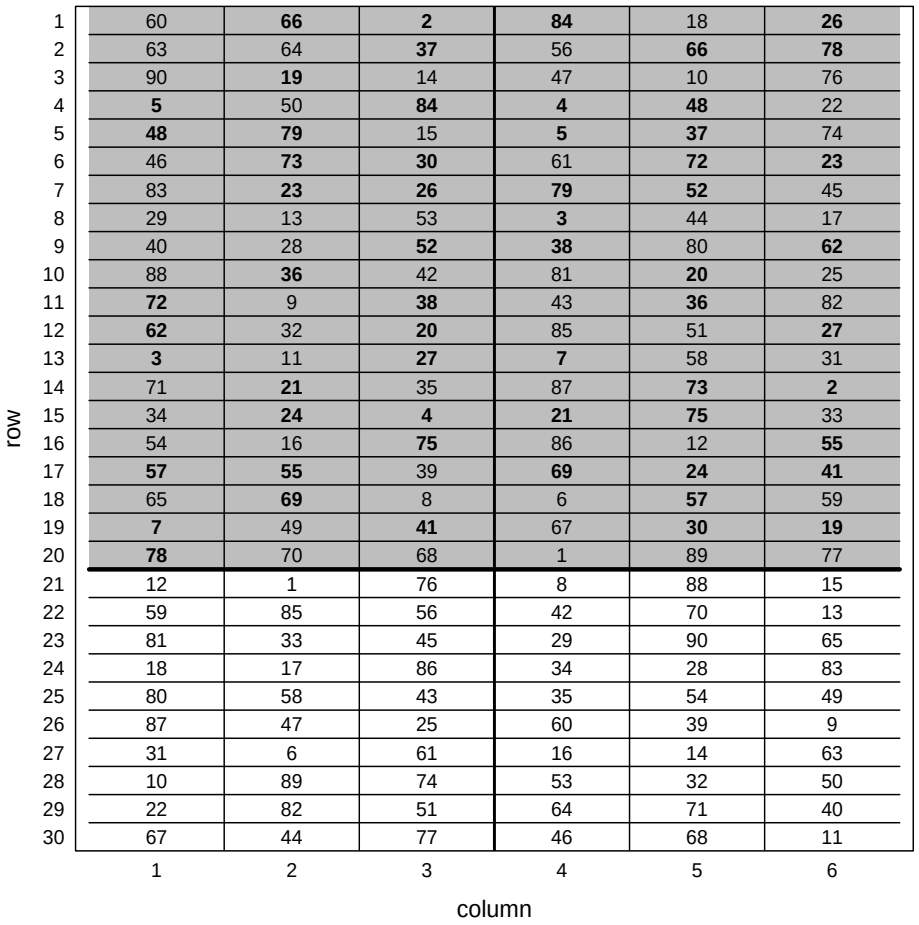


Figure 3: OT2 design for testing grain yield and oil content. Full layout for grain yield arranged as 30 rows by 6 columns with 2 replicates of 90 entries (block 1 = columns 1-3; block 2 = columns 4-6). Entry numbers are given for each plot. The subset of plots for testing oil content (shaded plots) comprises 2 replicates of 30 entries (numbers in bold face) and single replicates of 60 entries (numbers in plain face).

sizes 30 and 60, representing the replicated and unreplicated sets, respectively.

We note that there may be some loss in efficiency for grain yield in using an OT2 design, irrespective of the approach taken for its construction, compared with a design that is optimal for this trait alone. This will be investigated in the simulation study of Section 2.2.

2.1.2 Using mixtures of composite and individual plot samples

Recall that we have a scenario in which 30 of the 90 entries are being tested as individual replicates. Here the remaining 60 entries are tested as composite samples of grain from their (two) replicate plots. Thus we will be testing the same total number (120) of samples as in Section 2.1.1 but these are now a mixture of composite and individual plot samples.

In terms of the approaches for testing oil content as described in Section 2.1.1 we would use individual plot samples for two replicates of 30 entries (so for the layout in Figures

2 and 3 this corresponds to the 60 shaded plots with entry numbers in bold face) but instead of using individual plot samples for a single replicate of the remaining 60 entries (corresponding to the shaded plots with entry numbers in plain face) we composite the grain from these plots with their remaining replicate plot (that is, the non-shaded plots).

The use of a mixture of composite and individual plot samples has important implications for the statistical analysis. First we consider the analysis of a single trial. Using the notation of Section 1.1.1 we now assume for ease of explanation that the n_j plots comprise a fully replicated trial with b_j replicates of m varieties (so that $c_j r_j = n_j = m b_j$). Let m_s denote the number of varieties for which we will use composite samples. It is assumed that all replicates of each of these m_s varieties will be combined to form a composite sample so that the final number of grain samples to be quality tested is given by $s_j = (m - m_s)b_j + m_s$. Now we consider a transformation of the data vector $\boldsymbol{y}_j$ that is commensurate with a compositing process, namely the averaging of individual replicate data for a subset of the genotypes. This is a reasonable representation of the physical process when the sample units (in our case grain samples from individual replicates) enter into the composite in equal amounts and the physical act of pooling does not affect the trait of interest (Lancaster & Keller-McNulty, 1998). In this way the experimental units under consideration are 'transformed' from the original n_j plots to s_j samples. Of these s_j samples, m_s are composites of b_j plots and the remainder correspond to individual (original) plots. We denote the $s_j \times n_j$ transformation matrix by $\boldsymbol{D}_j$ and the transformed data by $\boldsymbol{z}_j = \boldsymbol{D}_j \boldsymbol{y}_j$. The spatial model for the data is then given by

$$\boldsymbol{z}_j = \boldsymbol{D}_j \boldsymbol{X}_j \boldsymbol{\tau}_j + \boldsymbol{D}_j \boldsymbol{Z}_{\boldsymbol{g}_j} \boldsymbol{u}_{\boldsymbol{g}_j} + \boldsymbol{D}_j \boldsymbol{Z}_{\boldsymbol{p}_j} \boldsymbol{u}_{\boldsymbol{p}_j} + \boldsymbol{D}_j \boldsymbol{e}_j \tag{4}$$

where the effects $\boldsymbol{\tau}_j$, $\boldsymbol{u}_{\boldsymbol{g}_j}$, $\boldsymbol{u}_{\boldsymbol{p}_j}$ and $\boldsymbol{e}_j$, design matrices $\boldsymbol{X}_j$, $\boldsymbol{Z}_{\boldsymbol{g}_j}$ and $\boldsymbol{Z}_{\boldsymbol{p}_j}$ and variance structures for $\boldsymbol{u}_{\boldsymbol{g}_j}$, $\boldsymbol{u}_{\boldsymbol{p}_j}$ and $\boldsymbol{e}_j$ are as defined in Section 1.1.1. This model involves some non-standard design matrices.

Finally we note that the MET analysis for data that involve a mixture of composite and individual plot samples can be obtained as an extension of the single site analysis in an analogous manner to Sections 1.1.1 and 1.1.2.

2.2 Accuracy of new approaches

The performance of the new approaches was assessed using a simulation study for the case described in Section 2.1, namely the MET with 7 trials each laid out as 30 rows by 6 columns and subsequent testing of 120 samples from each trial. Two series of field layouts were used corresponding to OT2 and OT1 designs. The same varieties were replicated in the two sets of designs. Data were generated according to the final model fitted to the real example of Section 1.2, that is, with genetic variance parameters as given in Table 3 and fixed effects and non-genetic variance parameters as given in Table 4.

Data were generated for the full trial layouts of 180 plots in each. For each generated data-set and design type, eight methods were used to obtain predicted variety effects. In the first seven methods a mixed model analysis was fitted and the E-BLUPs of variety

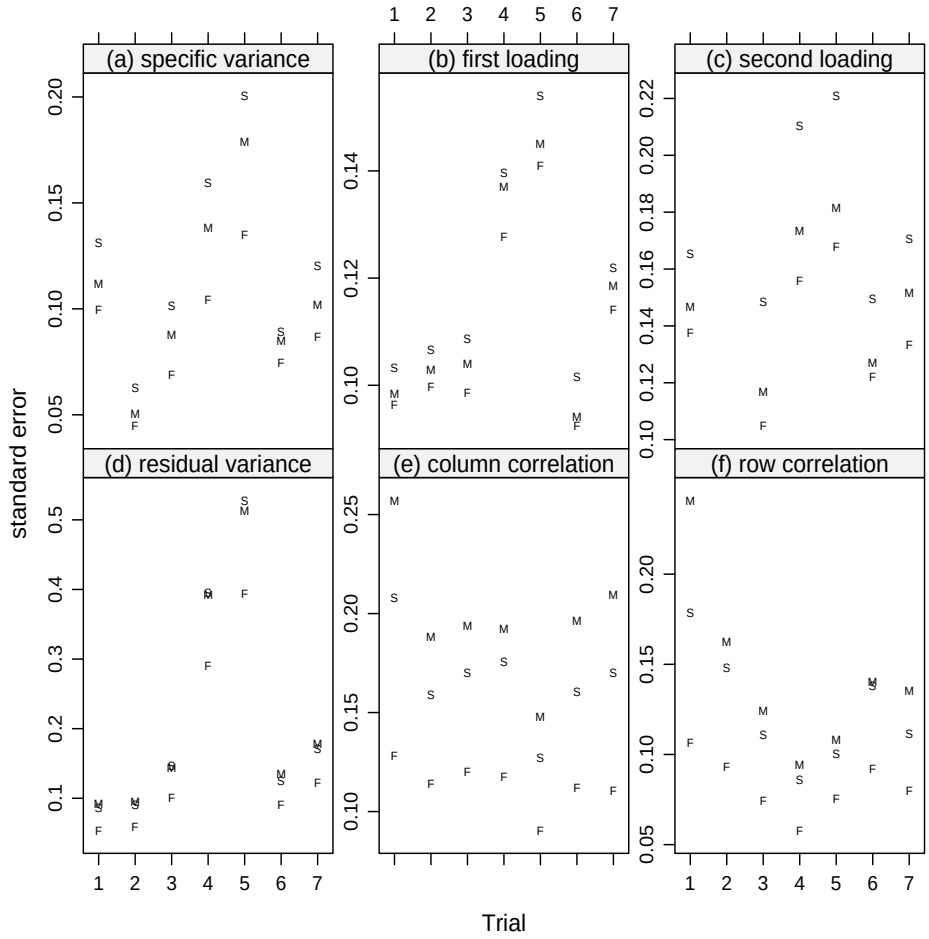


Figure 4: Simulation (N=300) standard errors for variance parameter estimates from models fitted to the full data-set (MF, labelled as "F"), the data corresponding to the subsetted plots (MS, labelled as "S") and the data comprising mixtures of composite and individual plot samples (MM, labelled as "M") for OT2 designs.

effects for each trial obtained. In the final method no analysis was conducted and the "predicted" variety effects were taken to be the raw data expressed as deviations from trial means. The methods were as follows:

MF: The true model was fitted to the full vector of generated data denoted by $\boldsymbol{y} = (\boldsymbol{y}_1' \ldots \boldsymbol{y}_7')'$ where $\boldsymbol{y}_j$ $(j = 1 \ldots 7)$ is the 180 $\times$1 data vector for the j^{th} trial

MFD: As for MF but a diagonal model for $\boldsymbol{G_e}$ was fitted

MS: The true model was fitted to the sub-vector of $\boldsymbol{y}$ corresponding to the subsetted plots for each trial (so data from 120 plots only)

MSD: As for MS but a diagonal model for $\boldsymbol{G_e}$ was fitted

MM: The true model was fitted to the vector $\boldsymbol{z} = (\boldsymbol{z}_1' \ldots \boldsymbol{z}_7')'$ where $\boldsymbol{z}_j = \boldsymbol{D}_j \boldsymbol{y}_j$ corresponds to the mixed sample type data (60 individual plot samples and 60 composite samples) for the j^{th} trial

Table 5: Correlations between true and predicted entry effects for diagonal models fitted to the full data-set (MFD), the data corresponding to the subsetted plots (MSD) and the data comprising mixtures of composite and individual plot samples (MMD). Correlations given for both OT2 designs (optimised for both grain yield and oil content) and OT1 designs (optimised for grain yield only). Values are averages over all entries in each trial.

	MFD		MSD		MMD	
Trial	OT2	OT1	OT2	OT1	OT2	OT1
1	0.927	0.926	0.874	0.871	0.912	0.910
2	0.946	0.945	0.900	0.894	0.926	0.927
3	0.924	0.923	0.867	0.857	0.899	0.897
4	0.957	0.957	0.911	0.904	0.927	0.930
5	0.866	0.866	0.780	0.767	0.823	0.825
6	0.924	0.926	0.864	0.859	0.897	0.899
7	0.938	0.937	0.888	0.881	0.918	0.917

MMD: As for MM but a diagonal model for $\boldsymbol{G_e}$ was fitted

MC: A model was fitted to the vector of data corresponding to a full compositing scheme (that is, 90 composite samples for each trial). In the absence of replication within-trial spatial variation can not be reliably modelled and the genetic variance model must be simplified. The 'best possible' sub-model of the true model for this scenario can be written as in equation (2) with the vector $\boldsymbol{\tau}$ comprising the trial means, the vector $\boldsymbol{u_p}$ omitted and the design matrix for the genetic effects given by an identity matrix. We may still apply an FA model to the data but the residual genetic effects ($\boldsymbol{\delta}$ in equation (3)) are completely confounded with the plot errors $\boldsymbol{e}$. Thus we write $\boldsymbol{u_g} = (\boldsymbol{\Lambda} \otimes \boldsymbol{I}_m)\boldsymbol{f}$ so that $\boldsymbol{G_e} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}' \otimes \boldsymbol{I}_{90}$ and assume $var\,(\boldsymbol{e}) = \boldsymbol{\Psi} \otimes \boldsymbol{I}_{90} = diag\,(\psi_j \boldsymbol{I}_{90})$.

MR: As for MC but no model was fitted

The correlations between the true and predicted variety effects were calculated for each method and design type. For the i^{th} variety in the j^{th} trial this was calculated (for any given method and design type) as

$$\frac{\sum_{k=1}^{N}\left(u_{g_{ijk}} - \bar{u}_{g_{ij.}}\right)\left(\tilde{u}_{g_{ijk}} - \bar{\tilde{u}}_{g_{ij.}}\right)}{\sqrt{\sum_{k=1}^{N}\left(u_{g_{ijk}} - \bar{u}_{g_{ij.}}\right)^2 \sum_{k=1}^{N}\left(\tilde{u}_{g_{ijk}} - \bar{\tilde{u}}_{g_{ij.}}\right)^2}} \tag{5}$$

where $u_{g_{ijk}}$ is the true (generated) effect for entry i in trial j and simulation k $(= 1 \ldots N)$ and $\tilde{u}_{g_{ijk}}$ is the associated predicted effect (E-BLUP from the analysis for the first seven methods and corrected raw data for the last). The term $\bar{u}_{g_{ij.}}$ denotes the mean across N simulations of the true effects for entry i in trial j and $\bar{\tilde{u}}_{g_{ij.}}$ is the analogous mean for the predicted effects. In our study $N = 300$ and corresponds to those simulations in which the algorithm for fitting the mixed models converged for *all* methods.

First we consider estimation of the variance parameters for those methods in which the full model was able to be fitted, namely MF, MS and MM. In terms of bias all of

Table 6: Correlations between true and predicted entry effects for diagonal models fitted to the data corresponding to the subsetted plots (MSD) and the data comprising mixtures of composite and individual plot samples (MMD) for OT2 designs only. Values are averages over all entries, over entries tested as two samples (replicates) and entries tested as single samples (replicates for MSD and composites for MMD).

	MSD			MMD		
Trial	all	2 samples	1 sample	all	2 samples	1 sample
1	0.874	0.917	0.853	0.912	0.908	0.914
2	0.900	0.934	0.883	0.926	0.925	0.926
3	0.867	0.912	0.845	0.899	0.900	0.899
4	0.911	0.941	0.897	0.927	0.931	0.925
5	0.780	0.851	0.745	0.823	0.831	0.819
6	0.864	0.912	0.840	0.897	0.903	0.895
7	0.888	0.927	0.868	0.918	0.920	0.916

these methods were very similar and showed no significant bias for any of the parameters. This was a particularly important finding for MM which involves spatial modelling of data derived from a mixture of composite and individual plot samples. We believe that this is the first occurrence in the literature of such an analysis. In terms of the precision of variance parameter estimates there were some interesting differences between methods. Figure 4 shows the simulation standard errors of variance parameter estimates for MF, MS and MM for OT2 designs. Note that the block and column variances have been omitted from this figure but showed little difference between methods. As expected the precision was best for MF for all parameters. The standard errors for genetic parameters (specific variances and loadings) were always lower for the method employing a mixture of composite and individual samples (MM) compared with the method using a subset of the plots (MS) whereas the converse was true for the spatial auto-correlation parameters. The latter may be expected due to spatial trend being "smeared" when grain samples from different plots are composited.

In terms of correlations between the true and predicted variety effects we first consider the diagonal models that correspond to separate analyses of individual trials. Table 5 gives the average correlations between true and predicted variety effects for each trial (ie. averaged over all the entries in a trial) for MFD, MSD and MMD for both design types. In terms of the effect of design types we immediately see that for the subsetting approach (MSD) there is a small but consistent increase in the correlations for OT2 compared with OT1 designs. This may have been expected due to the fact that the embedded designs for oil content were optimised in the OT2 scenario whereas in OT1 they were not. Importantly the correlations for the data corresponding to the full two replicate designs (MFD) were almost identical for the two design types. This implies that there will be minimal loss in efficiency for grain yield in using an OT2 design. Also there was no substantial or consistent difference between types when some plots were composited. We will therefore focus on OT2 designs in what follows.

Table 5 also shows that the correlations for the approach involving composite samples

Table 7: Correlations between true and predicted entry effects for factor analytic models fitted to the full data-set (MF for both OT2 and OT1 designs), the data corresponding to the subsetted plots (MS for both OT2 and OT1 designs), the data comprising mixtures of composite and individual plot samples (MM for both OT2 and OT1 designs) and the data comprising composite samples alone (MC). Also given are correlations for data comprising composite samples alone but with no model fitted (MR). Values are averages over all entries in each trial.

	MF		MS		MM		MC	MR
Trial	OT2	OT1	OT2	OT1	OT2	OT1	OT2	OT2
1	0.939	0.938	0.907	0.904	0.929	0.928	0.836	0.917
2	0.967	0.967	0.955	0.952	0.962	0.961	0.944	0.908
3	0.948	0.948	0.925	0.923	0.937	0.937	0.896	0.890
4	0.967	0.966	0.948	0.944	0.954	0.955	0.906	0.897
5	0.934	0.933	0.914	0.911	0.923	0.923	0.900	0.795
6	0.947	0.947	0.921	0.919	0.935	0.936	0.890	0.878
7	0.959	0.958	0.941	0.938	0.950	0.950	0.926	0.910

(MMD) were always superior to the subsetting approach (MSD). In these diagonal models the difference can largely be explained by the reduced replication in the latter. Table 6 shows the average correlations for each trial for the two classes of entry, namely the 30 tested as two separate samples (replicates) and the 60 tested as single samples (either single replicates in the case of MSD or composite of two replicates in the case of MMD). The correlations for the two replicate entries are lower for MMD compared with MSD. This may be due to the fact that the spatial correlation parameters are less well estimated with a mixture of composite and individual plot samples (see Figure 4). The correlations for the single sample entries are substantially higher for MMD compared with MSD reflecting the greater replication implicit in the composite samples. The overall result is that the average correlations for each trial are higher for MMD.

We now consider the correlations between true and predicted variety effects from the full MET analyses. Table 7 shows that the methods employing reduced replication, either with or without composite samples (MM and MS respectively) performed favourably when compared with the fully replicated method (MF). The MS and MM correlations were consistently lower than for MF but offset against these losses is the fact that $33\frac{1}{3}\%$ fewer samples were measured for MS and MM compared with MF so that the associated reduction in cost must be taken into account. The method based on mixtures of composite and individual plot samples (MM) was substantially better than the method based on reduced replication alone (MS) and there was no difference between the two types of design for MM whereas for MS the OT2 designs were superior. The full composite approach, either with (MC) or without (MR) a mixed model analysis had substantially lower correlations between true and predicted variety effects compared with all other methods.

We note that the genetic variances for oil content for the trials in this study were high relative to error and this is reflected in the high correlations between true and predicted variety effects. In our experience we have found that many quality traits exhibit

Table 8: Simulations based on reduced genetic variances and correlations. Correlations between true and predicted entry effects for factor analytic models fitted to the full dataset (MF for both OT2 and OT1 designs), the data corresponding to the subsetted plots (MS for both OT2 and OT1 designs), the data comprising mixtures of composite and individual plot samples (MM for both OT2 and OT1 designs) and the data comprising composite samples alone (MC). Also given are correlations for data comprising composite samples alone but with no model fitted (MR). Values are averages over all entries in each trial.

	MF		MS		MM		MC	MR
Trial	OT2	OT1	OT2	OT1	OT2	OT1	OT2	OT2
1	0.815	0.813	0.731	0.721	0.789	0.784	0.605	0.768
2	0.905	0.904	0.865	0.860	0.889	0.888	0.797	0.735
3	0.850	0.848	0.793	0.782	0.822	0.817	0.720	0.708
4	0.905	0.905	0.854	0.847	0.874	0.875	0.743	0.714
5	0.834	0.834	0.783	0.774	0.809	0.805	0.721	0.547
6	0.868	0.871	0.822	0.815	0.847	0.847	0.743	0.682
7	0.904	0.904	0.865	0.860	0.886	0.886	0.838	0.752

lower genetic variance and weaker genetic correlations than observed here. We therefore undertook a further simulation study in which we generated data based on the same non-genetic effects (that is, as in Table 4) but altered the genetic parameters so that the genetic variance for each trial was one quarter of the value used in the first simulation study. This was achieved by halving the specific variances and reducing the loadings. This not only reduced the genetic variances but also the genetic correlations, the average pairwise correlation being 0.69 compared with 0.82 for the first simulation study. The correlations between true and predicted variety effects from this simulation study for the full MET analyses are given in Table 8. This table reveals the same patterns as in Table 7 but the differences between methods are greater.

3 Using mixtures of composite and individual plot samples with multi-phase experiments

3.1 Background

Despite the importance of phenotyping quality traits in plant breeding programs and associated genomic research, literature on experimental design and statistical analysis for these traits is scarce. Most quality traits are obtained from multi-phase experiments in which entries (to represent varieties, breeding lines or other types of genetic material) are first grown in a field trial then further processed in the laboratory. Smith et al. (2006) presented a general mixed model approach for the analysis of multi-phase data, with particular emphasis on quality trait data that are often highly unbalanced and involve substantial sources of non-genetic variation and correlation. They also detailed is a new approach for experimental design that employs partial replication in all phases. The motivation for this was the high cost of obtaining quality trait data and thence the need to limit the total number of samples tested, but still allow use of the mixed model analysis. They conducted a simulation study to show that the combined use of the new designs and mixed model analysis has substantial benefits in terms of the genetic gain from selection.

3.2 Multi-phase experiments - statistical analysis

The approach we follow for the analysis of multi-phase plant breeding experiments builds on that of Brien (1983) and Wood et al. (1988). In both of these papers the analysis of multi-phase experiments is conducted by determining the experimental structure then including appropriate terms in an ANOVA table. The experiments considered by these authors are restrictive, however, in the sense that some degree of orthogonality is required. Brien (1983) discusses orthogonal designs that can be analysed using standard ANOVA. Wood et al. (1988) consider a class of two-phase designs with non-orthogonal block structure and provide an ANOVA approach to analysis but comment that a full analysis involving recovery of information would require a more sophisticated procedure based on REML estimation of variance parameters. Thus the linear mixed model provides a natural framework for the analysis of multi-phase experiments. We therefore use a general linear mixed model that removes any restrictions concerning orthogonality of block structures is proposed. In simple orthogonal cases the approach provides equivalent analyses to those proposed in Brien (1983) and Wood et al. (1988).

3.3 Multi-phase experiments - a hypothetical example

Before considering this general mixed model a simple orthogonal two-phase milling experiment that can be analysed using ANOVA is examined. It is assumed that r field replicates of g entries are grown in a field trial that is designed as an RCB. Grain samples

from each of the rg field plots are split into d smaller samples to be used as replicates in the laboratory process. Thus there is a total of $n = rgd$ samples to be milled. It is assumed that rg samples can be processed each day so that the full trial requires d days. Field plots are randomised to times in the milling process using an RCB design with days as blocks. A single sample from each of the rg field plots is processed each day and the plots are allocated completely at random within a day. The data measured for each sample is the flour yield.

In order to develop the analysis within an ANOVA framework the usual practice of assuming that block effects are random and treatment effects are fixed is followed. Thus, for illustrative purposes it is assumed here that entry effects are fixed. As previously noted the determination of the block structure can be difficult in the context of multi-phase experiments and the concepts in Brien (1983) may be helpful. The basic principle is to include terms in the model that capture the randomisation processes used in each phase of the experiment. In the two-phase quality experiment there are two randomisations, namely the randomisation of entries to field plots then the randomisation of field plots to 'positions' in the laboratory process. Thus the effects for block factors associated with each of these randomisations must be included in the analysis. Accordingly and based on the randomisation processes described above the symbolic model formula for the hypothetical example can be written as

$$\mathsf{y} \sim \underline{1} + \underline{\mathsf{entry}} + \mathsf{mrep} + \mathsf{frep} + \mathsf{frep.plot} + \mathsf{mrep.order} \tag{6}$$

where '1' represents an overall mean, entry is a factor with g levels, mrep is a factor (for replicates in the milling process) with d levels, frep is a factor (for field replicates) with r levels, plot is a factor (for plots within field replicates) with g levels and order is a factor (indexing the order of processing of samples within days) with rg levels. Thus the final term in (6) is the residual term that is also represented generically as units. Note the convention in the model formula of underlining to indicate those terms that correspond to fixed effects. The ANOVA table associated with the model in equation (6) is given in Table 9. A key feature of the analysis is the existence of a residual term for each of the two phases. The first phase (field plot) residuals are represented in the model by the term frep.plot and the second phase (laboratory) residuals by mrep.order (or units).

3.4 General linear mixed model for multi-phase experiments

The hypothetical milling example is atypical of trials for measuring quality trait data in that the data are rarely balanced nor are the designs orthogonal. Typically there is partial rather than complete replication in terms of both the field and laboratory. This has arisen from the work of Smith et al. (2006) who extended the idea of Cullis et al. (2006) who introduced so-called p-rep designs, by considering so-called pq-rep designs for two phase experiments, pqr-rep designs for three phase experiments and so on. Furthermore, as it the case with our examples, often only a subset of the entries (and/or plots) from the field trial is quality tested. Thus in general ANOVA cannot be used but a mixed model analysis

Table 9: ANOVA table for hypothetical milling example.

Strata/Decomposition	df	Model term
mean	1	1
mrep	$d-1$	mrep
mrep.order	$d(rg-1)$	
frep	$r-1$	frep
frep.plot	$r(g-1)$	
entry	$g-1$	entry
residual	$(r-1)(g-1)$	frep.plot
residual	$(d-1)(rg-1)$	units
total	rgd	

must be conducted. However, the same general principles are followed, in particular the inclusion of residual terms for all phases of the experiment.

As with the analysis of field trials, trend in multi-phase trial data can be modelled in order to improve the response to selection. In multi-phase quality trials the potential exists to model trend (spatial or temporal) associated with the residuals for any of the phases. The type of trend modelling depends on the trait and/or measurement process. We focus on an approach which is consistent with the analysis of the examples, ie flour yield and amylase in wheat. The modelling approach for field variation is analogous to that of Gilmour et al. (1997) in that trend (either field or laboratory) is partitioned into global trend, extraneous variation and local stationary trend. The latter is accommodated using covariance models. It is important to note that, in the spirit of a randomisation based analysis, terms in the mixed model that are associated with the block structure are maintained irrespective of their level of significance. In contrast, model-based terms and covariance structures are only included if found to be statistically significant.

For simplicity attention is restricted to two-phase experiments but the extension to more phases is straight-forward. An experiment is considered in which the first phase field trial consists of a rectangular array indexed by field rows $(1 \dots r)$ and columns $(1 \dots c)$ making a total of $n_p = rc$ plots. In the second phase laboratory trial it is assumed that a total of n samples is tested. Note that, like the field trial, many laboratory trials can be indexed using a two-dimensional co-ordinate system. For example in a wheat milling trial the samples may be milled as a set number, s, of samples per day for d days (so that $n = sd$). This has a two-dimensional spatial layout of m_r rows and m_c columns so often $n = m_r m_c$. Thus a rectangular structure for the laboratory process is assumed here. Extensions to non-rectangular or non-contiguous arrays for either the field or laboratory layouts are straightforward. The mixed model for the $n \times 1$ data vector $\boldsymbol{y}$ may be written as in a hierarchical manner by

$$\begin{aligned} \boldsymbol{y}|\boldsymbol{\beta}_f &= \boldsymbol{X}_m\boldsymbol{\tau}_m + \boldsymbol{Z}_f\boldsymbol{\beta}_f + \boldsymbol{Z}_{p_m}\boldsymbol{u}_{p_m} + \boldsymbol{e}_m \\ \boldsymbol{\beta}_f &= \boldsymbol{X}_f\boldsymbol{\tau}_f + \boldsymbol{Z}_g\boldsymbol{u}_g + \boldsymbol{Z}_{p_f}\boldsymbol{u}_{p_f} + \boldsymbol{e}_f \end{aligned} \tag{7}$$

where $\boldsymbol{\tau}_m$ and $\boldsymbol{\tau}_f$ are vectors of fixed effects for the laboratory and field respectively, $\boldsymbol{u}_{p_m}$ and $\boldsymbol{u}_{p_f}$ are vectors of peripheral effects for the milling and field respectively, $\boldsymbol{u}_g$ is the vector of entry effects and $\boldsymbol{e}_f$ and $\boldsymbol{e}_m$ are the vectors of residual effects for the milling and field respectively.

In the simplest case the fixed effects in (7) comprise a single effect, namely an overall mean, but may include effects for missing values or covariates to model trend for either the field and/or the milling phases. The vectors of peripheral effects include effects for those terms associated with the experimental design in each phase and other effects as required to model variation. The presence of two residual terms one for the first (field) phase and one for the second (milling) phase is fundamental. Note that not all field plots may be quality tested in which case the matrix $\boldsymbol{Z}_f$ will contain columns whose elements are all zero. In the next section we will incorporate this aspect in an explicit manner. Also note the assumption of random (rather than fixed) entry effects which is consistent with the applications we consider.

Variance models for the random effects in (7) are chosen from those described for the analysis of field trials and available in **ASReml-R** and **ASReml**.

As an illustration consider the ANOVA for the hypothetical milling experiment of Section 3.3 but now regard the entry effects as random rather than fixed. The vector $\boldsymbol{u}_g$ in equation (7) corresponds to the term entry in equation (6); $\boldsymbol{e}_f$ corresponds to frep.plot; $\boldsymbol{u}_{p_m}$ contains mrep effects, $\boldsymbol{u}_{p_f}$ contains frep effects; $\boldsymbol{e}_m$ corresponds to mrep.order and $\boldsymbol{\tau}_m$ contains a single parameter only, namely an overall mean. In the ANOVA model each set of random effects is assumed to be independent and each with the default variance matrix (a scaled identity matrix). Thus the general mixed model of equation (7) encompasses ANOVA models for multi-phase data (as discussed in Brien, 1983; Wood et al., 1988, for example) but has much broader application since non-orthogonal designs and unbalanced data are easily handled and more general covariance structures can be considered. The latter is particularly important for the modelling of local stationary trend associated with either the field or laboratory phase and extensions considered in the next section.

3.5 Multi-phase experiments with partial compositing

In this section we present an approach which combines the ideas found in Smith et al. (2011) and discussed in Section 2.1.2 with the approach presented above for the design of multi-phase experiments. For purpose of illustration and simplicity we consider the analysis of a two phase experiment.

The model is given by

$$\begin{aligned} \boldsymbol{y}|\boldsymbol{\beta}_f &= \boldsymbol{X}_m\boldsymbol{\tau}_m + \boldsymbol{Z}_f\boldsymbol{C}_f\boldsymbol{S}_f\boldsymbol{\beta}_f + \boldsymbol{Z}_{p_m}\boldsymbol{u}_{p_m} + \boldsymbol{e}_m \\ \boldsymbol{\beta}_f &= \boldsymbol{X}_f\boldsymbol{\tau}_f + \boldsymbol{Z}_g\boldsymbol{u}_g + \boldsymbol{Z}_{p_f}\boldsymbol{u}_{p_f} + \boldsymbol{e}_f \end{aligned} \tag{8}$$

where all of the vectors are as for (7). The matrices $\boldsymbol{C}_f$ and $\boldsymbol{S}_f$ are compositing and selection matrices which perform the compositing and selection of field plots. The elements

of $\boldsymbol{S}_f$ are either 0 or 1 while the elements of $\boldsymbol{C}_f$ are typically 0 or 0.5, but may be any other real number depending on the compositing regime.

For further details on the model and analysis see Cavanagh et al. (2012).

4 Examples

Both examples are kindly provided by Colin Cavanagh, Food Futures Flagship, CSIRO and involve phenotyping of lines (and commercial varieties) as part of the Multi-parent Advanced Generation Inter-cross (MAGIC) project see From mutations to MAGIC: resources for gene discovery & delivery in crop plants (2008) for details.

4.1 Narrabri Milling Experiment

The field trial was sown at Narrabri in northern NSW in 2010 and involved a total of 773 entries (760 experimental lines and 13 commercial varieties) sown in a resolvable p-rep design. The entire trial comprised 1000 plots arranged in a rectangular arrany with 50 column $\times$ 20 rows. Table 10 presents the frequency distribution for lines, commercial varieties and entries. Note that there was a sowing replacement with one experimental line sown in four plots instead of two plots. (This was not part of the original design.)

Table 10: Summary of frequency distribution for experiment lines, commercial varieties and entries in the Narrabri 2010 field trial.

Plots	Lines	Varieties	Entries
1	0	554	554
2	8	205	213
3	4	0	4
4	1	1	2
Total	13	760	773

A total of 480 entries was considered for milling. Of these 350 were selected on the basis of genetic diversity from the entire trial for those entries which did not get milled from a companion trial conducted at Yanco in 2009. An additional 130 entries was selected which were milled from the Yanco last year but maximize diversity of the current set. Table 11 presents a summary of the types of grain samples used in the final milling design. A total of 509 grain samples was created. These samples were classified according to the following status:

c1 a composite sample created from equal mixing of two field plots (of the same entry). There were 89 of these.

c2 a composite sample created from equal mixing of two field plots (of the same entry) but with sufficient grain to form two milling duplicates. There were 25 of these.

dup a sample of grain from a single field plot with sufficient grain to form two milling duplicates. These grain samples were taken from field plots with either a single replicate of an entry (10) or two replicates of an entry (2). For the latter case, the other plot was not used for milling. This ensured a more even distribution of replication across entries. There were 12 of these.

Table 11: Summary of type of grain samples used in the milling experiment for the Narrabri 2010 trial.

Sample Type	No.	Grain samples	Multipliers for Field plots	Entries	Mill samples
Composite (c1)	89	1	2	1	1
Comp-dup (c2)	25	1	2	1	2
Mill-dup (dup)	12	1	1	1	2
Field-rep (frep)	58	1	1	0.5	1
Field-single (sing)	325	1	1	1	1
Total		509	623	480	546

frep a sample of grain from each of the two field plots for each of 29 entries. There were 58 of these.

sing a sample of grain from a single field plot. There were 325 of these, representing the remaining entries which were not replicated in the field or chosen as a "dup" sample.

In summary, the milling design involved a total of 480 entries, 509 grain samples from 623 field plots and 546 milling samples. This strategy allowed for a more comprehensive sampling of the field trial, appropriate levels of direct replication for both phases (namely 6.0 and 7.3% for field and milling respectively). Additionally we have indirect replication of 28.4% of entries through composited samples. A total of 66 entries had either milling or direct field replication, and 89 indirect field replication (ie *c1* sample type).

The milling experiment design involved two resolvable blocks each of 39 days, with 7 samples per day.

Now comes the challenge to form an analysis!

4.1.1 Decomposition of sources of variation

Table 12 presents a skeletal decomposition of sources of variation for the Narrabri milling experiment. This is merely a convenient way to present the terms which have been fitted in the model to account for the randomisation processes, as well as accounting for potential sources of variation. Note that the design search for the field, milling and plate phases usually includes all of these terms.

4.2 Yanco amylase experiment

The field trial was sown at Yanco, southern NSW in 2009 and is a companion to the Narrabri trial. A total of 1077 entries was sown in a p-rep design. The trial comprised 1620 plots arranged in a rectangular array with 20 columns by 81 rows. There were three

Table 12: Decomposition of sources of variation for Narrabri milling experiment.

Decomposition	Model term
mean	1
mrep	MRep
mrep.mday	MRep : MDay
mrep.mday.mord	
frep	FRep
fcol	FCol
frow	FRow
fcol.frow	
entry	Entry
residual	FCol : FRow
residual	units

adjoining (physical) bays such that bay 1 spanned all columns and rows 1 to 27, bay 2 spanned all columns and rows 28 to 54 and bay 3 spanned all columns and rows 55 to 81.

Table 13: Summary of frequency distribution for experiment lines, commercial varieties amd entries in the Yanco 2009 field trial.

Plots	Lines	Varieties	Entries
1	0	556	556
2	0	507	507
3	9	0	9
4	4	0	4
7	1	0	1
Total	14	1063	1077

Table 13 presents the frequency distribution for lines, commercial varieties and entries in the field trial. Note that the 14 commercial varieties had additional plots, as is often the case in these trials. The trait we consider is the amylase content of the flour. This is measured using an ELISA plate in which samples of flour are placed in wells on a 96 well plate (8 columns by 12 rows). This enzymatic reaction produces amylase which is then quantified by measuring the absorbance by passing each plate through a scanner.

This experiment is a three phase experiment. Phase 1 is the field trial, phase 2 is the milling trial and phase 3 is the ELISA plate assay. In considering an approach to the design, consideration needs to be given to all potential sources of variation in all three phases. Additionally we need to determine the experimental unit and the observational unit for this experiment. The experimental unit is the field plot, while the observational unit is the ELISA plate well. There may be sources of variation due to the scanner (aligned with the rows and columns of the plate), as well as variation between ELISA plates, variation due to milling days, milling order, milling order within milling days from

Table 14: Summary of frequency distribution of replication for each phase of the Yanco amylase experiment

Phase	Sample type	Replication 1	2	3	Samples	Units
ELISA		488	132	0	620	752 (16)
Milling						
	Comp	131	0	0	131	131
	Other	390	48	1	439	489
	Total	521	48	1	570	620
Field						
	Total				698	570

phase 2, and finally variation between field bays, field columns, field rows and field plots from phase 1.

A total of 498 entries was selected to be tested, using a similar approach outlined above (based on maximising genetic diversity). Grain samples were classified as for the Narrabri milling experiment, except there was no *c2* types. (This idea has only recently been thought of and tested.) A total of 570 grain samples were used for milling. These comprised 131 *c1* samples, 132 *frep* samples, 49 *dup* samples and 258 *sing* samples. This information is presented in table 14. Of the 49 samples classified as *dup*, 48 had 2 milling duplicates, while 1 was triplicated in the milling phase. This occurred during the experiment, as the technician replaced another grain sample with this sample.

The resulting milling experimental design had a total of 620 milling units, comprising 62 days with 10 samples per day. The milling design was a resolvable q-rep design with 2 resolvable MillReps being MillDays 1 to 31 and 32 to 62 respectively.

Table 15: Summary of frequency distribution of milling replicate and ELISA plate of the Yanco amylase experiment

Plate	MillRep-1	MillRep-2
1	94	0
2	94	0
3	94	0
4	78	16
5	9	85
6	1	93
7	0	94
8	16	78

The ELISA phase of the experiment was a r-rep chain-block design with a total of 8 ELISA plates, comprising 96 wells arranged in 8 PlateColumns $\times$ 12 PlateRows. A total of 132 of the 620 milling units were duplicated in this phase. An additional 16 blanks were

Table 16: Decomposition of sources of variation for Yanco amylase experiment.

Decomposition	Model term
mean	1
plate	Plate
pcol	PColumn
prow	Prow
plate.pcol	Plate : PColumn
plate.prow	Plate : PRow
plate.prow.prow	
mrep	MRep
mrep.mday	MRep : MDay
mrep.mday.mord	
fbay	FBay
fcol	FCol
frow	FRow
fcol.frow	
entry	Entry
residual	FRol : FRow
residual	MRep : MDay : MOrd
residual	units

included (two per plate) for quality control purposes. These data were excluded from the analysis.

Use of a chain-block design was necessary due to the need to commence processing the flour samples before the full milling experiment was completed. Fortunately, some flour samples could be frozen which allowed linkage of the milling days with the ELISA plates. The frequency distribution of MillRep $\times$ Plate is presented in table 15, illustrating the nature of the chain-block design with the linkage between MillDays 1 to 5 (aka MillRep) and Plates.

The resulting frequency distribution of entries was in the ELISA experiment was $(254, 235, 8, 1)$ for 1, 2,3 and 4 wells respectively.

4.2.1 Decomposition of sources of variation

Table 16 presents a skeletal decomposition of sources of variation for the Narrabri milling experiment. This is merely a convenient way to present the terms which have been fitted in the model to account for the randomisation processes, as well as accounting for potential sources of variation. Note that the design search for the field, milling and plate phases usually includes all of these terms.

Bibliography

Brien, C. J. (1983). Analysis of variance tables based on experimental structure. *Biometrics* 39 53–59.

Cavanagh, C., Smith, A., Butler, D., Thompson, R. & Cullis, B. (2012). On the analysis of multi-phase experiments with partial compositing. *submitted* xx xx.

Cullis, B., Smith, A. & Coombes, N. (2006). On the design of early generation variety trials with correlated data. *Journal of Agricultural, Biological and Environmental Statistics* 11 381–393.

Cullis, B. R. & Gleeson, A. C. (1991). Spatial analysis of field experiments - an extension to two dimensions. *Biometrics* 47 1449–1460.

Eccleston, J. & Chan, B. (1998). Design algorithms for correlated data. In *COMPSTAT98 Proceedings in Computational Statistics*. Heidelberg: Physica-Verlag, 41–52.

From mutations to MAGIC: resources for gene discovery, v. & delivery in crop plants (2008). Cavanagh, c. and morell, m.k. and mackay, i. and powell, w. *Current Opinion in Plant Biology* 11 215–221.

Gentle, J. E. (1998). *Numerical linear algebra for applications in statistics.* Statistics and Computing. New York: Springer-Verlag.

Gilmour, A. R., Cullis, B. R. & Verbyla, A. P. (1997). Accounting for natural and extraneous variation in the analysis of field experiments. *Journal of Agricultural, Biological and Environmental Statistics* 2 269–293.

Lancaster, V. A. & Keller-McNulty, S. (1998). A review of composite sampling methods. *Journal of the American Statistical Association* 93 1216–1230.

Martin, R. J. & Eccleston, J. (1997). Construction of optimal and near-optimal designs for dependent observations using simulated annealing. Research report 479/97, Dept. of Probability and Statistics, University of Sheffield.

Oakey, H., Verbyla, A., Cullis, B., Wei, X. & Pitchford, W. (2007). Joint modelling of additive and non-additive (genetic line) effects in multi-environment trials. *Theoretical and Applied Genetics* 114 1319–1332.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Smith, A., Cullis, B. R. & Thompson, R. (2001). Analyzing variety by environment data using multiplicative mixed models and adjustments for spatial field trend. *Biometrics* 57 1138–1147.

Smith, A., Lim, P. & Cullis, B. (2006). On the design of multi-phase experiments for quality trait data. *Journal of Agricultural Science* 144 393–409.

Smith, A., Thompson, R., Butler, D. & Cullis, B. (2011). The analysis of variety trials using mixtures of composite and individual plot samples. *Journal of the Royal Statistical Society, Series C* 60 437–455.

Stefanova, K., Smith, A. & Cullis, B. (2009). Enhanced diagnostics for the spatial analysis of field trials. *Journal of Agricultural, Biological and Environmental Statistics* 14 1–19.

Wood, J. T., Williams, E. R. & Speed, T. P. (1988). Non-orthogonal block structure in two-phase designs. *Australian Journal of Statistics* 30A 225–237.

Chapter

D

OD: An optimal design companion to ASReml

1 Introduction

Experimental designs for plant improvement studies are often selected on an informal basis, with due consideration to desirable properties such as replication, blocking, orthogonality or connectedness. In practice, adjustments are sometimes made to experimental procedures if an *off the shelf* design does not quite fit the intended application. Optimal design allows flexible combinations of the design specification parameters for special cases, and prespecified residual or treatment correlation structures.

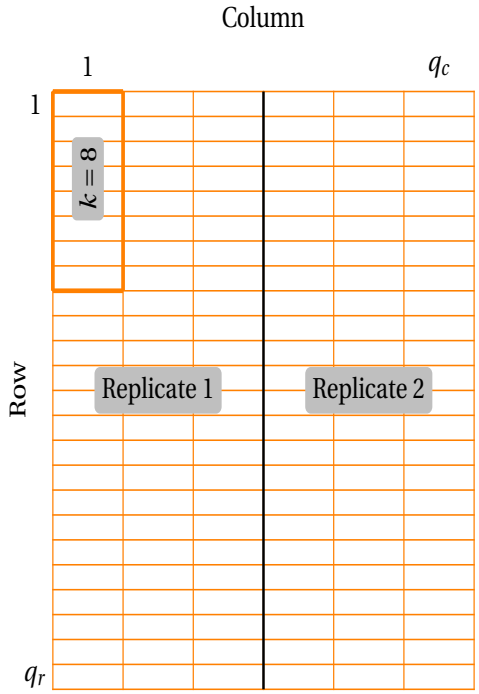
Table 1 defines some notation used here for the design parameters, and the physical configuration of experiments typically encountered in field, glasshouse or laboratory evaluations. Field, or even glasshouse, experiments are typically arranged in a regular *row* $\times$ *column* grid of experimental units, with resolvable replicates aligned with either rows or columns (Figure 1). Field experiments are managed mechanically, with sowing and harvesting operations along rows, and operations such as plot trimming within columns.

Table 1: Symbols used to describe the configuration of genetic evaluation experiments.

Symbol	Definition
v	Number of treatments (genotypes) in the experiment
k	Number of experimental units per (incomplete) block
s	Number of blocks per replicate
r	Number of resolvable replicates
r_i	Number of replicates of treatments $i = 1, \ldots, v$
q_r	Number of rows of experimental units in the physical layout
q_c	Number of columns of experimental units in the physical layout
N	Total number, $q_r \times q_c$, of experimental units

Sound experimental design promotes the statistically efficient separation of genetic and environmental effects in the analysis of phenotypic observations. Commonly, designs used in plant breeding programs fall within the general class of block designs, and include

Figure 1: Example layout of a replicated field trial with $q_r = 24$, $q_c = 6$, $v = 72$, $r = 2$, $s = 9$ and $k = 8$. Resolvable replicates are aligned with columns.



complete or incomplete block, row-column and α-designs (Williams & John, 1996). Exceptions include large scale early generation evaluation, where methods based on Yates (1936a), Federer & Raghavarao (1975a) and Lin & Poushinsky (1983) are in use; more recently partially replicated designs based on Cullis et al. (2006a) are chosen in this situation. Design methods that extend to spatially correlated data have been of recent interest (Butler et al., 2008; Cullis et al., 2006a; Williams et al., 2006; Chan, 1999; Martin & Eccleston, 1997; Martin, 1986), largely motivated by advances in fitting the linear mixed model in this context (Cullis et al., 1998; Gilmour et al., 1997; Cullis & Gleeson, 1991). Few theoretical results exist, and numerical optimization has been almost exclusively used in the construction of A-optimal designs (Coombes, 2002; Chan, 1999; Eccleston & Whitaker, 1999; Martin & Eccleston, 1997). Unreplicated field trials, or those with restricted replication, follow the same grid pattern as Figure 1, and some of the parameters k, s, and r may still retain their meaning in special cases such as augmented or partially replicated designs.

2 Introduction to optimal design

An optimal design can be simply defined as an experimental design that is *optimal* with respect to some statistical criterion, $\mathcal{O}$, say, based on some measure of the treatment information. The goal is to maximise the (treatment) information available to the practitioner for the statistical model under investigation, for a given number of experimental units,

and sampling regime. Optimal design provides a flexible framework for efficient design in novel experimental situations, such as where no design exists in the literature for the particular combination of treatments, replication and blocking structure, or the underlying statistical model precludes a standard design. The theory of optimal experimental design is credited to Kiefer (1959) and is followed by numerous overviews including the works by Silvey (1980), Atkinson & Donev (1992), and Atkinson et al. (2007).

Consider the simple linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}$$

where $\boldsymbol{y}$ is a vector of observations, $\boldsymbol{X}$ is a $N \times t$ design matrix (assumed full rank), and $\boldsymbol{e}$ is a N vector of residuals assumed normally distributed with mean zero and variance σ^2. For simplicity, initially let the t columns of $\boldsymbol{X}$ be t continuous explanatory variates, and let $\mathcal{D}$ denote the design region from which the rows of $\boldsymbol{X}$, $\boldsymbol{x}_{i\cdot}$, can be drawn. In the discussion that follows, $\boldsymbol{x}_{i\cdot}$ may, depending on context, refer to a row of the design matrix $\boldsymbol{X}$, or the vector of predictor values at design point i, where there are n distinct design points.

The *information* available to the experimenter is quantified in terms of the variances of the estimators $\hat{\boldsymbol{\beta}}$. The variance-covariance matrices of estimable functions of the $\hat{\boldsymbol{\beta}}$ are formed from the inverse information matrix $\boldsymbol{C}^{-1}$ where

$$\begin{aligned} \boldsymbol{C} &= \sigma^{-2}(\boldsymbol{X}^T\boldsymbol{X}), \text{ and} \\ \text{var}(\hat{\boldsymbol{\beta}}) &= \boldsymbol{C}^{-1} = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1} \end{aligned}$$

Also, the variance of the predicted values

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$$

is

$$\text{var}(\hat{\boldsymbol{y}}) = \sigma^2\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$$

In general, optimal design seeks to choose an $\boldsymbol{X}$ that minimizes some function of $\text{var}(\hat{\boldsymbol{\beta}})$, that is, $\mathcal{O} = \mathcal{F}(\boldsymbol{C}^{-1})$ in this illustration. Forms for $\mathcal{F}$ are usually from the so-called *alphabet* series of criteria, some of which are encapsulated by Kiefer (1974) in the Φ_k family.

At this point, it is useful to introduce two optimality criteria that illustrate at least one consequence of the general equivalence theorem outlined below. Of these, a much studied criterion is D-optimality, where $|\boldsymbol{C}^{-1}|$ is minimized, or equivalently $|\boldsymbol{C}|$ is maximized. D-optimality minimizes the generalized variance of $\hat{\boldsymbol{\beta}}$. A criterion concerned with the variance of the predicted response is G-optimality, which seeks to minimize the maximum variance of the predicted response over $\mathcal{D}$. That is, G-optimality is defined as $\min(\max(\text{var}(\hat{\boldsymbol{y}})))$.

2.1 Continuous and exact designs

A design is specified by an initially arbitrary measure ξ assigning unit mass over the region $\mathcal{D}$ (Cox & Reid, 2000).

$$\xi = \left\{ \begin{array}{cc} x_1 & x_2 \dots x_n \\ w_1 & w_2 \dots w_n \end{array} \right\} \tag{1}$$

where the $\{x_i\}$ are the n distinct design points and $\{w_i\}$ are the associated weights. As ξ is a measure, $\int_{\mathcal{D}} \xi(dx) = 1$, and $0 \leq w_i \leq 1$, and $\sum_i w_i = 1$. Denoting the information matrix for $\boldsymbol{\beta}$ in models such as those in equation (2) by $\boldsymbol{M}(\xi)$, then

$$\boldsymbol{M}(\xi) = \int_{\mathcal{D}} \boldsymbol{x}_{i\cdot} \boldsymbol{x}_{i\cdot}^T \xi(dx) = \sum_{i=1}^{n} w_i \boldsymbol{M}(\bar{\xi}_i) \tag{2}$$

where

$$\boldsymbol{X}^T \boldsymbol{X} = \sum_{i=1}^{N} \boldsymbol{x}_{i\cdot} \boldsymbol{x}_{i\cdot}^T \tag{3}$$

and the measure $\bar{\xi}_i$ assigns unit mass at design point i.

Mathematically, the problem of finding an optimal design is approached in two ways (Goos, 2002):

Continuous where the number of observations at a design point may be non-integral. Design construction can be viewed as purely a numerical optimization problem (Wu, 1978), starting from the general equivalence theorem (below).

Discrete where the design respects the practicality of the integral constraint. A measure that refers to a design realizable in integers is written as ξ_N and the weights $\{w_i\}$ in expression (1) are such that $w_i N = r_i$, where r_i is the number of replicates at design point i, and $\sum r_i = N$. The summation form of (2)

$$\boldsymbol{M}(\xi_N) = \sum_{i=1}^{n} w_i \boldsymbol{x}_{i\cdot} \boldsymbol{x}_{i\cdot}^T$$

for exact designs becomes the *normalised* version of (3), that is, $\boldsymbol{M}(\xi_N) = \boldsymbol{X}^T \boldsymbol{X}/N$.

Let $\Psi(\boldsymbol{M})$ represent a general optimality criterion to be optimized. In the theory of optimal continuous design, the General Equivalence Theorem (Kiefer & Wolfowitz, 1960) states

1. the optimal design ξ^* minimizes Ψ
2. the minimum of the derivative of Ψ (Ψ') evaluated in ξ^* is ≥ 0
3. Ψ' evaluated in ξ^* achieves its minimum at the design points

Table 2: Optimality criteria

Criteria	Definition	Computation	Interpretation
A	$\sum_{i=1}^{p}(1/\lambda_i)$	$\text{tr}\left[\boldsymbol{M}^{-1}\right]$	Minimize the variance of elementary treatment contrasts.
D	$\prod_{i=1}^{p}\lambda_i$	$\lvert\boldsymbol{M}\rvert$	Minimize the variance of parameter estimates.
E	$\max_{i=1,\ldots,p}(1/\lambda_i)$		Minimize the variance of the least well determined contrast (of a set).
G	λ		Minimize the maximum variance of predicted values.
V	λ		Minimize the average variance of predicted values.

The theorem assumes that Ψ is convex and differentiable, and as a result any minimum found will be global. *D*-optimality, where $\Psi(\boldsymbol{M}(\xi)) = -\log(|\boldsymbol{M}(\xi)|)$, and $\Psi' = p - \text{var}(\hat{\boldsymbol{y}})$, satisfies these conditions. A direct consequence is that *D*-optimal designs are also *G*-optimal, in that the maximum of the prediction variance over $\mathcal{D}$ is minimized (Goos, 2002).

D-optimality and *G*-optimality are only two of many that have been proposed, others found useful in practice include:

A-optimality, which minimizes the average of the $\text{var}(\hat{\boldsymbol{\beta}})$

E-optimality, which minimizes the variance of the least well estimated contrast $\boldsymbol{a}^T\boldsymbol{\beta}$ given $\boldsymbol{a}^T\boldsymbol{a} = 1$, and

V-optimality, which minimizes the average prediction variance over $\mathcal{D}$.

For each of these criteria there is an analogue to the General Equivalence Theorem, compactly described in terms of the general criterion $\max \Phi(\boldsymbol{M}(\xi))$, where Φ is a concave function on the set of information matrices $\boldsymbol{M}(\xi)$. For *D*-optimality Φ is the concave function $\log|\boldsymbol{M}(\xi)|$, for example (Atkinson, 1982).

Kiefer (1974) expressed these optimality measures in terms of the sum of the k^{th} powers of the (non-zero) eigenvalues $\{\lambda_i\}$ of $\boldsymbol{M}$. Assuming $\boldsymbol{M}$ is of full rank, the general Φ_k value is $(\sum \lambda^{-k}/p)^{1/k}$ for $0 \leq k \leq \infty$. Special cases of interest are $k = 0, 1, \infty$, corresponding to *D*-, *A*-, and *E*-optimality, respectively. Table 2 summarises the common criteria found in the literature and introduced in this section.

In practice all designs are discrete (or *exact*), however, the General Equivalence Theorem is not valid in general for discrete designs. Efficient, though not necessarily optimal exact designs may sometimes be found by rounding $N \times w_i$, from a corresponding optimal continuous design, to the nearest integer. The optimality of continuous designs can be proven, whereas a computer search is generally necessary to establish the optimality of a discrete design. It should be bourne in mind that when comparing exact designs, it is the

relative values of the optimality criterion Ψ that are of importance, rather than how well ξ_N approximates ξ^* (Atkinson et al., 2007).

2.2 Categorical designs

The simple example introduced in (2) assumed that the design matrix $\boldsymbol{X}$ was of full rank and the columns of $\boldsymbol{X}$ were continuous variates, such as successive powers in a polynomial regression, for example. This is an example of a response surface type design, where in general the objective is to select or exchange the rows of $\boldsymbol{X}$ from the design space $\mathcal{D}$ to optimize Ψ. The emphasis in pioneering work from agriculture such as Fisher (1935) was on evaluating the effect of qualitative factors on biological response variables, as well as their inclusion as a means of accounting for extraneous environmental variation. The theory of optimal design is often presented in terms of response surface designs, however, the principles and results hold in both settings, though the computational processes may differ.

2.3 Scope

We only consider discrete, categorical designs where typically, though not exclusively, the columns of $\boldsymbol{X}$ are dummy variables formed from categorical factors. In this context, a *design* can be considered as a permutation $\mathcal{P}$ of the rows of $\boldsymbol{X}$, with $\mathcal{D}$ then the set of all such permutations.

In plant genetic evaluation programs, all treatment comparisons are generally of interest. The A-optimality criterion is equivalent to minimising the average parwise variance of elementary contrasts and is usually considered the most appropriate in these circumstances. We use the A-optimality criterion, or a random effects equivalent, exclusively here. Exceptions, though not considered here, arise when the goal could be to maximise genetic gain, for example, in which case alternative criteria may be more appropriate.

3 The linear mixed model

This section revisits the general linear mixed model introduced in Chapter 1, and defines some elements relevent in the context of design, and the examples that follow. As before, we begin where a vector of observations, $\boldsymbol{y}$, is modelled by a linear combination of explanatory variables. For example, grain yield from plots in a genotype evaluation trial may be modelled by a linear predictor comprising fixed and random terms representing management effects, random genetic effects, blocking effects (such as field *row* or *column*), for example, and a possibly correlated residual error structure. If grain samples from these plots were subsequently milled in the laboratory for flour yield, then the linear predictor would be extended to include milling effects such as day, trend within day, machine or operator effects in addition to the environmental effects carried from the field.

Historically, elements of the random error, and indeed those in all random terms, were usually considered uncorrelated. This assumption is relaxed here using optimal design methods, and a range of variance models may be considered for all random terms. Typically, first order autoregressive models are often assumed for the error structure arising in a field experiment, though many other models are possible. Gilmour et al. (1997) discuss a recommended approach to the analysis such data.

In genetic improvement experiments, the genotype effects are often considered as realisations of a random variable, sometimes with a simple variance model that assumes these effects are independent and identically distributed with a common variance, σ_g^2, say. Plant breeders keep crossing records, so the *pedigree* of an individual and its relatives is known, and an additive genetic relationship matrix ($\boldsymbol{A}$), and its inverse can be derived (see Henderson, 1976). More recently, DNA markers can be used to derive a genetic relationship matrix among individuals that is *identical in state*; either way, both relax the *iid* assumption and allow a more plausible correlation structure among individuals.

With $\boldsymbol{y}$ as the $N \times 1$ vector of observations, the linear mixed model can be written as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e} \tag{4}$$

where $\boldsymbol{\tau}$ is a $t \times 1$ vector of fixed effects, $\boldsymbol{X}$ is an $N \times t$ design matrix which associates observations with the appropriate combination of fixed effects, $\boldsymbol{u}$ is the $b \times 1$ vector of random effects, $\boldsymbol{Z}$ is the $N \times b$ design matrix which associates observations with the appropriate combination of random effects, and $\boldsymbol{e}$ is the $N \times 1$ vector of residual errors. Note that $\boldsymbol{X}$ may not necessarily be of full column rank.

We assume

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{e} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \sigma^2 \begin{bmatrix} \boldsymbol{G}(\boldsymbol{\gamma}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}(\boldsymbol{\phi}) \end{bmatrix} \right) \tag{5}$$

where the matrices $\boldsymbol{G}$ and $\boldsymbol{R}$ are functions of parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$, respectively. The parameter σ is an overall scale parameter, and without loss of generality in the design context we can set $\sigma = 1$. We subsequently use $\boldsymbol{\kappa}$ to denote the vector of variance parameters in place of $\boldsymbol{\gamma}$, which is traditionally reserved for variance ratios.

3.1 Variance structures for the errors: $\boldsymbol{R}$

The vector $\boldsymbol{e}$ will in some situations be a series of vectors indexed by a factor or factors. For example, in the classical analysis of a single field experiment $\boldsymbol{R} = \sigma^2 I_n$, however, for a series of s such experiments (Smith et al., 1998, for example) we may consider $\boldsymbol{R}$ to have the form $\oplus_{j=1}^{s} \sigma_j^2 I_{n_j}$. That is, we can write $\boldsymbol{e} = [\boldsymbol{e}_1', \boldsymbol{e}_2', \ldots, \boldsymbol{e}_s']'$ and the $\boldsymbol{e}_j$ represent the errors of *sections* of the data. More generally, Cullis et al. (1998) consider the spatial analysis of multi-environment trials in which

$$\begin{aligned} \boldsymbol{R}_j &= \boldsymbol{R}_j(\boldsymbol{\phi}_j) \\ &= \sigma_j^2(\boldsymbol{\Sigma}_j(\boldsymbol{\rho}_j) + \psi_j \boldsymbol{I}_{n_j}) \end{aligned} \tag{6}$$

and each *section* represents an independent experiment. This model accounts for between trial error variance heterogeneity (σ_j^2), possibly a different spatial variance model for each trial ($\boldsymbol{\Sigma}_j$ and $\boldsymbol{\rho}_j$) and different measurement errors (ψ_j). As in Chapter 1, we assume a separable first order autoregressive process across the *column* and *row* dimensions, so for experiment j

$$\boldsymbol{\Sigma}_j(\boldsymbol{\rho}_j) = \boldsymbol{\Sigma}_{cj}(\rho_{cj}) \otimes \boldsymbol{\Sigma}_{rj}(\rho_{rj})$$

where ρ_{cj} and ρ_{rj} are the scalar autocorrelation parameters. We only consider single experiments in this Chapter and so can drop the subscript j.

3.2 Variance structures for the random effects: $\boldsymbol{G}$

The $b \times 1$ vector of random effects may be composed of w subvectors $\boldsymbol{u} = [\boldsymbol{u}_1^T \ \boldsymbol{u}_2^T \ \ldots \ \boldsymbol{u}_w^T]^T$ where the subvectors $\boldsymbol{u}_i$ are of length b_i and these subvectors are usually assumed independent normally distributed with variance matrices $\boldsymbol{G}_i$. We can write $\boldsymbol{G}$ as

$$\boldsymbol{G} = \oplus_{i=1}^{w} \boldsymbol{G}_i = \begin{bmatrix} \boldsymbol{G}_1 & 0 & \ldots & 0 & 0 \\ 0 & \boldsymbol{G}_2 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & \boldsymbol{G}_{w-1} & 0 \\ 0 & 0 & \ldots & 0 & \boldsymbol{G}_w \end{bmatrix}$$

There is a corresponding partition in $\boldsymbol{Z}$: $\boldsymbol{Z} = [\boldsymbol{Z}_1 \ \boldsymbol{Z}_2 \ \ldots \ \boldsymbol{Z}_w]$. Each submatrix, $\boldsymbol{G}_i$, is assumed to be the kronecker product of one, two or more component matrices. These matrices are indexed for each of the factors constituting the term in the linear model. For example, in a multi-trial analysis the term *site:genotype* has two factors and so the matrix $\boldsymbol{G}_i$ is comprised of two component matrices defining the variance structure for each factor in the term.

Assuming separability, models for the component matrices $\boldsymbol{G}_i$ are defined by

$$\boldsymbol{G}_i \ = \ \boldsymbol{G}_{i1} \otimes \boldsymbol{G}_{i2} \otimes \ldots \otimes \boldsymbol{G}_{if}$$

for f component factors. The vector $\boldsymbol{u}_i$ is therefore assumed to be the vector representation of a f-way array. For two factors ($f = 2$) the vector $\boldsymbol{u}_i$ is simply the vec() of a matrix with rows and columns indexed by the component factors in the term, where vec() of a matrix is an operator which stacks the columns of its argument into a vector.

3.3 Variance models

Variance models for $\boldsymbol{R}$ and $\boldsymbol{G}$ can be either on the correlation or covariance scale. The simplest correlation model is the identity model: for example, if the v genotype effects in a plant variety experiment were considered normally distributed as $N(0, \sigma_g^2)$, then that component matrix of $\boldsymbol{G}$ would be a scaled identity $\sigma_g^2 \boldsymbol{I}_v$. We consider other variance models for random genetic effects below.

3.4 Estimation

The mixed model equations (Robinson, 1991) which are given by

$$\begin{bmatrix} \boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{Z} \\ \boldsymbol{Z}^T\boldsymbol{R}^{-1}\boldsymbol{X} & \boldsymbol{Z}^T\boldsymbol{R}^{-1}\boldsymbol{Z}+\boldsymbol{G}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\tau}} \\ \tilde{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{y} \\ \boldsymbol{Z}^T\boldsymbol{R}^{-1}\boldsymbol{y} \end{bmatrix}. \tag{7}$$

These can be written as

$$\boldsymbol{C}\tilde{\boldsymbol{\beta}} = \boldsymbol{W}^T\boldsymbol{R}^{-1}\boldsymbol{y} \tag{8}$$

where $\boldsymbol{C} = \boldsymbol{W}^T\boldsymbol{R}^{-1}\boldsymbol{W} + \boldsymbol{G}^*$, $\boldsymbol{W} = [\boldsymbol{X} \;\; \boldsymbol{Z}]$, $\boldsymbol{\beta} = \left[\boldsymbol{\tau}^T \;\; \boldsymbol{u}^T\right]^T$ and

$$\boldsymbol{G}^* = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}^{-1} \end{bmatrix}.$$

Also note that

$$\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta} = \begin{bmatrix} \hat{\boldsymbol{\tau}} - \boldsymbol{\tau} \\ \tilde{\boldsymbol{u}} - \boldsymbol{u} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \; \boldsymbol{C}^{-1} \right) \tag{9}$$

3.5 Random treatments, partitioning and decomposing genetic effects

In many applications the treatments, usually genetic lines, are considerd as random effects. In this case we partition $\boldsymbol{u}$ as $(\boldsymbol{u}_g^T, \boldsymbol{u}_b^T)^T$ where the genotype effects are in $\boldsymbol{u}_g$ and other non-genetic random effects are in $\boldsymbol{u}_b$, and assume:

$$\begin{bmatrix} \boldsymbol{u}_g \\ \boldsymbol{u}_b \\ \boldsymbol{e} \end{bmatrix} = N\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \sigma_g^2\boldsymbol{G}_g & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_b(\boldsymbol{\kappa}_b) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}(\boldsymbol{\phi}) \end{bmatrix} \right)$$

where $\boldsymbol{\kappa}_b$ and $\boldsymbol{\phi}$ are vectors of unknown variance components and correlation parameters. The expanded mixed model equations are now:

$$\begin{bmatrix} \boldsymbol{Z}_g^T\boldsymbol{R}^{-1}\boldsymbol{Z}_g + \boldsymbol{G}_g^{-1} & & \\ \boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{Z}_g & \boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{X} & \\ \boldsymbol{Z}_b^T\boldsymbol{R}^{-1}\boldsymbol{Z}_g & \boldsymbol{Z}_b^T\boldsymbol{R}^{-1}\boldsymbol{X} & \boldsymbol{Z}_b^T\boldsymbol{R}^{-1}\boldsymbol{Z}_b + \boldsymbol{G}_b^{-1}) \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}}_g \\ \hat{\boldsymbol{\tau}} \\ \tilde{\boldsymbol{u}}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z}_g^T\boldsymbol{R}^{-1}\boldsymbol{y} \\ \boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{y} \\ \boldsymbol{Z}_b^T\boldsymbol{R}^{-1}\boldsymbol{y} \end{bmatrix}$$

where $\boldsymbol{Z} = [\boldsymbol{Z}_g \; \boldsymbol{Z}_b]$. Sweeping out those equations for $\boldsymbol{\tau}$ and $\boldsymbol{u}_b$ gives

$$\begin{aligned} \left(\boldsymbol{Z}_g^T\boldsymbol{S}\boldsymbol{Z}_g + \boldsymbol{G}_g^{-1}\right)\tilde{\boldsymbol{u}}_g &= \boldsymbol{Z}_g^T\boldsymbol{S}\boldsymbol{y} \\ \boldsymbol{C}_{gg}\tilde{\boldsymbol{u}}_g &= \boldsymbol{Z}_g^T\boldsymbol{S}\boldsymbol{y} \end{aligned}$$

where

$$\boldsymbol{S} = \boldsymbol{R}^{-1} - \boldsymbol{R}^{-1}[\boldsymbol{X}\boldsymbol{Z}_b] \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{X} & \cdots \\ \boldsymbol{Z}_b^T\boldsymbol{R}^{-1}\boldsymbol{X} & \boldsymbol{Z}_b^T\boldsymbol{R}^{-1}\boldsymbol{Z}_b + \boldsymbol{G}_b^{-1} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{X}^T \\ \boldsymbol{Z}_b^T\boldsymbol{R}^{-1} \end{bmatrix},$$

$$\mathsf{var}(\tilde{\boldsymbol{u}}_g - \boldsymbol{u}_g) = \boldsymbol{C}^{gg} \tag{10}$$

and $\boldsymbol{C}^{gg} = \boldsymbol{C}_{gg}^{-1}$. This suggests an optimality criterion for design based on some $\mathcal{F}(\boldsymbol{C}^{gg})$.

The assumption above that $\text{var}(\boldsymbol{u}_g) = \sigma_g^2 \boldsymbol{I}_v$ may be plausible in certain cases where genotypes are from diverse genetic backgrounds, and the degree of relatedness among individuals is insignificant. Typically, in genetic improvement programs there is shared ancestry among individuals and the simple model of independent genotype effects is unrealistic.

A first step in decomposing the genetic effects is to distinguish between *additive* (that is, the combined effects of alleles at two or more gene loci are equal to the sum of their individual effects) and *non-additive* effects. Assuming additive and non-additive effects are independent, we can decompose $\boldsymbol{u}_g$ as

$$\boldsymbol{u}_g = \boldsymbol{u}_a + \boldsymbol{u}_{\bar{a}} \tag{11}$$

with $\text{var}(\boldsymbol{u}_g) = \sigma_a^2 \boldsymbol{A} + \sigma_{\bar{a}}^2 \boldsymbol{I}_v$, where $\boldsymbol{u}_a$ is the vector of additive genetic effects, and $\boldsymbol{u}_{\bar{a}}$ is the vector of non-additive genetic effects, both associated with the design matrix $\boldsymbol{Z}_g$

The matrix $\boldsymbol{A}$ is a known (fixed) additive genetic relationship matrix derived from ancestral records, however, from the mixed model equations (7) it is actually $\boldsymbol{A}^{-1}$ that is required. Numerous methods exist, with many dating back to Henderson (1976), to generate $\boldsymbol{A}^{-1}$ directly from pedigree records. The decomposition in (11) can be extended to include other genetic relationships, such as dominance, but this will not be considered any further here.

Often $\dim(\boldsymbol{A}) > v$, that is there are more individuals in the pedigree than will actually appear in the design. On that basis, the (complete) vector of genotype effects, $\boldsymbol{u}_G$, say, could be partitioned as $\boldsymbol{u}_G^T = (\boldsymbol{u}_g^T, \boldsymbol{u}_{\bar{g}}^T)^T$, where $\boldsymbol{u}_{\bar{g}}$ represent those related genotypes that do not appear in the design, but, may be used in calculating $\mathcal{O}$ in certain situations. Butler et al. (2012) discuss this further, but it will not be expanded on in this document.

The implications for design of recognising a genetic covariance structure among treatments are at least twofold:

1. The optimum allocation of genotypes to experimental units will be affected by the relatedness of neighbours,

2. The optimal choice of test genotypes to replicate in series of *p-rep* designs may be better determined.

4 Optimal design criteria

Table 2 lists several optimality criteria that have found popular use. As previously noted, the A-optimality criterion is equivalent to minimising the average pairwise variance of all

elementary treatment contrasts; it is usually considered the most appropriate in circumstances such as comparative genetic evaluation experiments where all treatments are of equal interest.

4.1 Fixed effects

Assuming genotypes are fixed effects, consider the mixed model equations in (7) reordered (that is, permute the columns of $\boldsymbol{W}$ and the corresponding elements of $\boldsymbol{\beta}$), so that the information matrix $\boldsymbol{C}$ in (8) and its generalised inverse are partitioned as

$$C = \begin{bmatrix} \boldsymbol{C}_{gg} & \boldsymbol{C}_{go} \\ \boldsymbol{C}_{og} & \boldsymbol{C}_{oo} \end{bmatrix}; \; C^{-} = \begin{bmatrix} \boldsymbol{C}^{gg} & \boldsymbol{C}^{go} \\ \boldsymbol{C}^{og} & \boldsymbol{C}^{oo} \end{bmatrix}; \tag{12}$$

where $\boldsymbol{C}_{gg}$ is the $v \times v$ coefficients matrix for genotypes, and $\boldsymbol{C}_{oo}$ is the coefficients matrix for all other fixed and random effects. If $\boldsymbol{\tau}_g$ is the $v \times 1$ (sub)vector of fixed genotype effects in $\boldsymbol{\tau}$, from (9) we can write

$$\text{var}(\hat{\boldsymbol{\tau}}_g - \boldsymbol{\tau}_g) = \boldsymbol{C}^{gg} \tag{13}$$

The *A-optimality* criterion minimises the average variance of the elementary contrasts $(\tau_i - \tau_j)$, and

$$\text{var}(\hat{\tau}_i - \hat{\tau}_j) = \text{v}_{ij} = c^{ii} + c^{jj} - 2c^{ij}$$

where $\boldsymbol{C}^{gg} = \{c^{ij}\}$. Denoting the *A-value* of a design by $\mathcal{A} = \bar{\text{v}}$, it can be shown Raghavarao (1971), for example, that

$$\mathcal{A} = \frac{2}{v-1}\left(\text{tr}\,[\boldsymbol{C}^{gg}] - \frac{1}{v}\mathbf{1}_v^T \boldsymbol{C}^{gg}\mathbf{1}_v\right) \tag{14}$$

and a design is said to be *A-optimal* if the *A-value* is a minimum for all designs under consideration

$$\mathcal{A}_{opt} = \min_{\mathcal{D}}\{\mathcal{A}_i\}.$$

If $\boldsymbol{C}^{-}$ is a Moore-Penrose generalised inverse, $\sum_{ij} c^{ij} = 0$ and the *A-value* of a design is just

$$\mathcal{A} \propto \text{tr}\,[\boldsymbol{C}^{gg}]$$

4.2 Random effects

If genotypes are considered as random effects, possibly with a known relationship matrix (that is, $\text{var}(\boldsymbol{u}_g) = \sigma_g^2 \boldsymbol{A}$), an *A-value* analagous to (14) can be derived from (7) in a similar manner to the above. Bueno Filho & Gilmour (2003), for example, use this direct random effects equivalent of A-optimality in assessing competing designs for various treatment correlation structures. In such cases, that is, models that include a known genetic relationship matrix, an alternative criterion can be derived from considering the so-called *coefficient of determination*, or the squared correlation between $\boldsymbol{u}_g$ and $\tilde{\boldsymbol{u}}_g$. It

turns out that this leads to proposing the average prediction error variance (PEV) as an appropriate criterion, that is

$$\mathcal{O} \propto \mathrm{tr}\,[\boldsymbol{C}^{gg}]$$

using (10).

5 Algorithms for optimal design

Algorithms for optimal design have historically been considered in the context of designs with quanitative factors, for both continuous and exact cases. Early work in this area pre-dates the widespread use of high speed computing, and borrows from generic methods for function minimisation. In adapting these early methods to categorical designs, the *objective function* is replaced by the chosen criterion, such as A- or D-optimality, or PEV, for example, and the *interchange* algorithms are recast in a permutation framework.

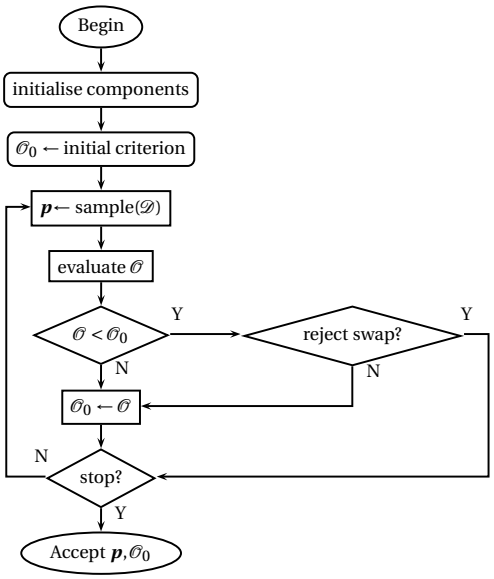
Even for small designs it is generally impossible to enumerate all solutions in $\mathcal{D}$, and an efficient sampling strategy directed by an optimisation algorithm is necessary. Design search methods share a common set of features in exploring the design space $\mathcal{D}$, and these include:

1. A suitable starting design
2. An optimality criterion $\mathcal{O}$
3. An interchange policy to move to a neighbouring configuration
4. Rules to assess the effect of a treatment interchange on $\mathcal{O}$
5. An acceptance policy for new configurations
6. A stopping rule to terminate the search

Figure 2 illustrates the simplified structure of a design optimization algorithm. The key features underlying this in the context of the computing paradigm discussed below include:

- Specifying the linear model and the pre-specified random components.
- Choosing an appropriate criterion $\mathcal{O}$.
- A mechanism to identify the target treatment factor and an interchange policy.
- An efficient optimization strategy to search the design space.

Figure 2: A simplified design optimization procedure.



5.1 Search algorithms

The *reject* and *stop* decision boxes in Figure 2 together with the sampling of $\mathcal{D}$ encapsulate the search algorithm in the optimization process. Rejection rules may be probability based, in which case inferior solutions will sometimes be accepted in order to widen the sampling of $\mathcal{D}$. Stopping rules may be based on iteration count, a minimum threshold for a change in the criterion, or an analagous physical process. Two search algorithms have been implemented in `od()`, and these are described below.

5.1.1 Simulated annealing

Simulated annealing is a stochastic search method introduced by Metropolis et al. (1953). Kirkpatrick et al. (1983) proposed an algorithm based on the analogy between the annealing of solids and solving combinatorial optimisation problems. Using this analogy, the Boltzmann probability

$$P(E) \propto e^{[-E/kt]} \tag{15}$$

determines the probability distribution of system energies E for a given temperature t. The constant k is Boltzmann's constant. The energy E corresponds to the value of the objective function and the *states* of the solid correspond to feasible solutions.

Within a sequence of iterations, a new solution E_2 is accepted unconditionally if $\Delta_E = E_2 - E_1$ is negative, otherwise E_2 is accepted according to Metropolis's criterion based on the Boltzmann probability. That is, a random number δ in $[0, 1]$ is generated and E_2 accepted if $\delta \leq \exp(-\Delta_E/t)$.

The minimum energy state corresponds to the optimum solution and several cooling schedules have been proposed in the literature Osman (1991); Lundy (1985).

Simulated annealing has been used in optimal design problems (Chan, 1999; Elliot et al., 1999) but Coombes (2002) found its performance inferior to tabu search based methods.

5.1.2 Tabu search

The tabu algorithm Glover (1989, 1990) is an iterative procedure characterised by flexible memory structures and local search strategies, with the ability to escape local minima. A basic tabu algorithm involves three main strategies: forbidding, freeing and short-term strategy.

The forbidding strategy guards against cycling problems by preventing solutions visited within the last T_s iterations from being revisited. T_s is normally called the *tabu* list size and the tabu list is a FIFO queue. The freeing strategy is used in conjunction with an aspiration criterion to decide what (if any) candidates exit the tabu list. The short-term strategy manages the interaction between the two previous strategies. For example, a local descent strategy could be used to evaluate *admissible* solutions in the neighbourhood of the current solution.

The work of Coombes (2002) suggests that reactive tabu search (Battiti & Tecchiolli, 1994) could be more efficient as the optimisation loop in design search problems. Reactive tabu modifies the basic strategy to include a tabu list of dynamic length, determined by a self tuning heuristic.

`od()` implements the tabu search strategy operating with a memory resident search list (Woodruff & Zemel, 1993). A unique key is generated for each design permutation vector $\boldsymbol{p}_i$, and stored in a rapid access memory structure. The design key is calculated as a 32 bit checksum using the computing industry standard Cyclic Redundancy Check (CRC) algorithm (Press et al., 1996) with polynomial 0x04C11DB7.

6 A design generator: od()

Many of the concepts in the previous sections of this chapter are being incorporated into an R package `od()`, where designs are specified using symbolic model formulae in the spirit of ASReml. We begin with some brief computing details specific to `od()` that put the function arguments in context.

6.1 Computational background

Let $\boldsymbol{W} = [\boldsymbol{X}\ \boldsymbol{Z}_g\ \boldsymbol{Z}_b]$ and permute the columns of $\boldsymbol{W}$ such that $\boldsymbol{W} = [\boldsymbol{W}_1\ \boldsymbol{W}_2]$ and $\boldsymbol{W}_1 = \boldsymbol{Z}_g$ (or that partition of $\boldsymbol{Z}_g$ conformal with the subset of $\boldsymbol{u}_g$ forming $\mathcal{O}$).

Given invariant design matrices for non-genetic effects, our goal is to minimise $\mathcal{O}$ through an optimal allocation of genotypes to the experimental units. This is conveniently achieved with a permutation matrix $\boldsymbol{P}$ operating on the rows of $\boldsymbol{W}_1$. Initially $\boldsymbol{P} = \boldsymbol{I}_N$. Consider a perturbation function $S()$ operating on the rows of $\boldsymbol{P}$, such that $\boldsymbol{P}^* = S(\boldsymbol{P})$

is a new permutation. A new design is produced as $\boldsymbol{W}_1^* = \boldsymbol{P}^*\boldsymbol{W}_1$. Computationally, a design is represented in a permutation vector $\boldsymbol{p}$ that indexes the rows of $\boldsymbol{P}$, and a new design $\boldsymbol{p}^*$ is generated as $\boldsymbol{p}^* = s(\boldsymbol{p})$, where $s()$ is the corresponding vector perturbation function. A straightforward interchange function for $s()$ is typically chosen which swaps two elements of $\boldsymbol{p}$(rows of $\boldsymbol{P}$), subject to any resolvability constraints. Given an initial $\boldsymbol{p}_0 = [1, 2, \ldots, n]$, the algorithm minimises $\mathcal{O}$ through repeated applications of $s(\boldsymbol{p})$ under the supervision of an optimisation strategy such as TABU search or simulated annealing (see Coombes, 2002). At the time of writing the TABU implementation was unreliable and simulated annealing is used in the examples that follow.

The optimization proceeds as shown in Figure 2, where a move to an new (neighbouring) design is achieved using the permutation strategy describe above, and $\mathcal{O}$ is calculated as:

1. Form the mixed model equations

   $\boldsymbol{W}^T\boldsymbol{R}^{-1}\boldsymbol{W} + \boldsymbol{G}^*$ where $\boldsymbol{G}^* = \boldsymbol{G}^{-1} \oplus \mathbf{0}$

2. Absorb the equations for the effects in $\boldsymbol{W}_2$ to form $\boldsymbol{C}_{gg}$

3. Form $\boldsymbol{C}^{gg}$ and calculate $\mathcal{O}$

6.2 The od()function

Table 3 outlines the arguments to od() and the corresponding components of the linear model or methodology to which they refer (where applicable). Section 9 reproduces the R help pages for the od() package.

Table 3: Arguments to od()

Argument	Value	Model/method context
fixed	$\sim$ *fixed model terms*	$\boldsymbol{X}$
random	$\sim$ *random model terms*	$\boldsymbol{Z}$
rcov	$\sim$ *formula* specifying the error terms	$\boldsymbol{R}$
permute	$\sim$ *formula* specifying term(s) to permute	$\boldsymbol{W}_1$
swap	$\sim$ *formula* controlling legal treatment exchanges	$S()$
Gstart	*list* assigning values of pre-specified variance parameters	$\boldsymbol{G}$
Rstart	*list* assigning values of pre-specified variance parameters	$\boldsymbol{R}$
ginverse	*list* specifying inverse relationship matrices	$\boldsymbol{A}^{-1}$
method	c('avalue','PEV')	$\mathcal{O}$
search	c('anneal', 'tabu', 'random')	
data	*dataframe* containing initial configuration	

Variance models for random terms are specified as special model functions in the same manner as ASReml-R. Currently these models are limited to id(), idv(), ar1(), ped() and ide() while the package is being developed. A data frame in which to resolve the named terms in the model formulae, and containing the initial design configuration, must be

provided; this is in the usual *observation* × *variate* array with model terms declared as factors or variables as appropriate.

The function returns an object of class od with the following components:

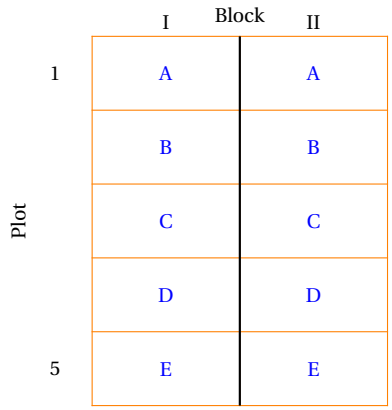
- A dataframe with the *permute* terms in design order
- The permutation vector $\boldsymbol{p}$.
- The optimality criterion $\mathcal{O}$.
- List(s) of the pre-specified variance components.
- The function call.

Methods for summary() and plot() exist, though currently have limited capability.

7 Examples

7.1 A randomised block

We begin by illustrating some features of od() with the simplest and arguably most commonly used block design. Consider the following experimental layout:



Configuration parameters (Table 1)

$v = 5$
$k = 5$
$s = 1$
$r = 2$
$q_r = 5$
$q_c = 2$
$N = 10$

```
R> ## Initial configuration
R> rb <- data.frame(Trt=rep(LETTERS[1:5],2), Block=factor(rep(1:2,each=5)),
+     Column=factor(rep(1:2,each=5)), Row=factor(rep(1:5,2)))
R> ## generate a design
R> rb.od <- od(fixed= ~Trt, random= ~Block, permute= ~Trt, swap= ~Block,
+     maxit=100, data=rb)
R> names(rb.od)

  [1] "call"        "design"      "criterion"   "permutation" "Gstart"
  [6] "Rstart"
```

Before considering the returned object *rb.od*, there are some implicit and explicit actions of the call that warrant explanation:

- *Block* appears in the random formula but no variance model or initial values have been given. The default model in this case is a scaled identity, with a default scale factor, that is, the variance component for blocks, of 0.1.
- The *rcov* model formula is not specified, so the default error model, $N(\mathbf{0}, \boldsymbol{I}_{10})$ has been used, remembering that σ^2 is fixed at 1.0.
- The *swap* model formula enforces the resolvability constraints of the design by restricting treatment exchanges to occur only within levels of the *Block* factor.

The optimized design is returned in the *design* component of the returned object, and the corresponding permutation vector $\boldsymbol{p}$ is returned in the *permutation* component.

```
rb.od$permutation
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```
rb.od$design
```

```
   Trt Block Column Row units
1    A     1      1   1     1
2    B     1      1   2     2
3    C     1      1   3     3
4    D     1      1   4     4
5    E     1      1   5     5
6    A     2      2   1     6
7    B     2      2   2     7
8    C     2      2   3     8
9    D     2      2   4     9
10   E     2      2   5    10
```

The (apparently) optimized design is just a copy of the initial dataframe, a result that surprises new users. This example resolves into complete blocks, has no incomplete blocking, extraneous effects or correlated error structure to drive treatment exchange. Under this model all configurations are optimal and *manual* randomization of treatments is the only guard against systematic effects.

To illustrate this point, an autoregressive error structure is added to the model and initial values are set using the *od.init()* function.

```
R> rb.init <- od.init(Block=0.1, "Column:Row|Column"=0.2, "Column:Row|Row"=0.4)
R> rbsp.od <- od(fixed= ~Trt, random= ~ Block, rcov= ~ar1(Column):ar1(Row),
+     permute= ~Trt, swap= ~Block, Rstart=rb.init, maxit=100, data=rb)
R> rbsp.od$design
```

```
   Trt Block Column Row
1    A     1      1   1
2    C     1      1   2
3    E     1      1   3
4    B     1      1   4
5    D     1      1   5
6    E     2      2   1
7    A     2      2   2
8    D     2      2   3
9    C     2      2   4
10   B     2      2   5
```

The initial A-value is 0.9320, while that of the optimized design above is 0.6300. The final configuration exhibits some classic properties of a spatial design based on an autoregressive process, such as diagonal self-neighbours and no self-adjacencies; that is the design is binary with respect to the *Row* factor:

```
R> table(rbsp.od$design$Trt,rbsp.od$design$Row)
```

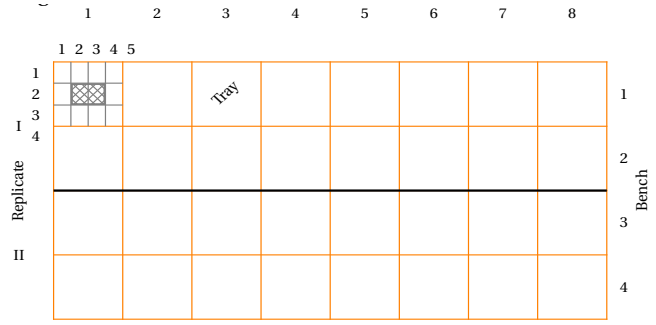
```
  1 2 3 4 5
A 1 1 0 0 0
B 0 0 0 1 1
C 0 1 0 1 0
D 0 0 1 0 1
E 1 0 1 0 0
```

In this small example, the binary configuration will most likely always be found, but as N gets larger there is no guarantee that a binary design will result. However, if the initial design is binary then the optimal design will also be binary, subject to an appropriate model.

7.2 A glasshouse example

This example is a glasshouse experiment from the national LMA screening program (Mrva & Mares, 2001). Briefly, this is stage one of a multi-stage process where (wheat) plants are grown in pots in a glasshouse prior to being transferred to other laboratory procedures. Two replicates of the treatments are arranged in pots in a 3×4 grid within trays across glasshouse benches, where trays and benches are contiguous. The centre two pots in each tray are left empty (blank), and are indicated by cross-hatching in the figure below. The main purpose of this example is to further illustrate the action of the *swap* argument.

```
R> ## LMA glasshouse winter 2011
R> ## design factors, no treats
R> ## 448 unique treats (+extra stds/rep)
```



$v = 448$
$r = 2$
$k = 12$
$s = 46$
$q_r = 12$
$q_c = 92$

```
R> ## At 10 per tray, 46 trays/rep, 460 units/rep
R> ## 92 cols X 12 rows
R>
R> Column <-  rep(1:92,rep(12,92))
R> Row <-  rep(1:12,92)
R> Replicate <- rep(rep(1:2,c(6,6)),92)
R> Tray <- matrix(rep(rep(1:92,rep(3,92))),nrow=23,byrow=TRUE)
R> Tray <- as.vector(t(cbind(Tray,Tray,Tray,Tray)))
R> ## Blank - control in centre two cells of each tray
R> cc <- sort(c(seq(from=2,by=4,length.out=23),seq(from=3,by=4,length.out=23)))
R> rr <- c(2,5,8,11)
R> Blank <- rep(0,92*12)
R> Blank[(is.element(Column,cc) & is.element(Row,rr))] <- 1
R> Treat <- rep(0,92*12)
R>
R> ghex <- data.frame(Column=factor(Column), Row=factor(Row),
+     Replicate=factor(Replicate), Tray=factor(Tray), Blank=factor(Blank),
+     Treat=Treat)
R> ghex[1:15,]
```

	Column	Row	Replicate	Tray	Blank	Treat
1	1	1	1	1	0	0
2	1	2	1	1	0	0
3	1	3	1	1	0	0
4	1	4	1	2	0	0
5	1	5	1	2	0	0
6	1	6	1	2	0	0
7	1	7	2	3	0	0
8	1	8	2	3	0	0
9	1	9	2	3	0	0
10	1	10	2	4	0	0
11	1	11	2	4	0	0
12	1	12	2	4	0	0
13	2	1	1	1	0	0

```
14        2   2         1    1     1     0
15        2   3         1    1     0     0
```

At this point the design factors exist but the treatments are yet to be allocated; this process is not relevent to the discussion and has been omitted for brevity. Note that the factor *Blank* identifies two *zones*: the empty pot positions, and the rest. The call to generate a design using the complete dataframe *gh* is:

```
R> gh.od <- od(fixed = ~ Entry, random = ~ Replicate+Tray+Column+Row,
+    rcov = ~ ar1(Column):ar1(Row), permute = ~ Entry, swap = ~ Replicate:Blank,
+    Gstart=gh.init, Rstart=gh.init, maxit=5, data=gh)
```

where the main purpose of the spatial error structure is to promote some neighbour balance properties in the configuration. The *swap* argument in this example limits treatment exchanges to be within the intersection of the levels of the nominated factors (*Replicate:Blank*), thus maintaining resolvability and banning swaps between zones.

```
R> table(ghex$Replicate,ghex$Blank)
```

```
     0   1
1 460  92
2 460  92
```

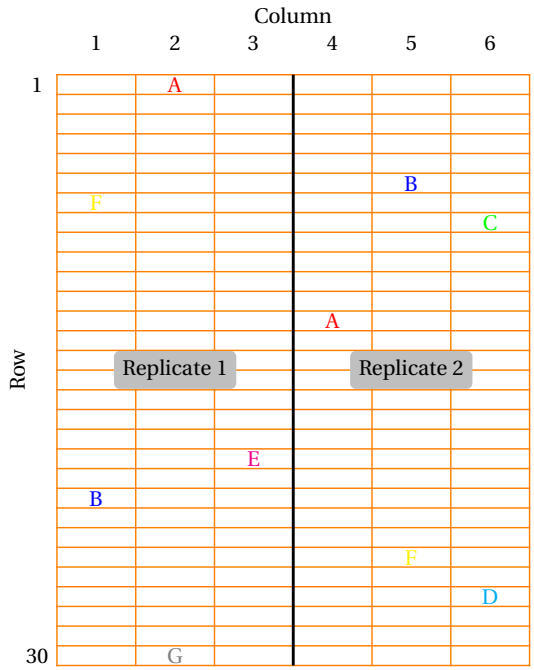
Perhaps not the most efficient solution given the wasted legal swaps within the empty zones, but nontheless effective. An alternative strategy would have been to omit the blank units from the dataframe and, if considered desirable, use a power model for the spatial component (should one have been implemented).

7.3 Partially replicated designs

As the name suggests, *p-rep* designs (Cullis et al., 2006b) replicate a proportion p of the test treatments in resolvable replicates with the remaining treatments appearing once only. The figure below illustrates this property.

```
R> id <-rep(LETTERS[1:7],c(2,2,2,1,1,1,1))
R> prep <- data.frame(Trt=as.vector(
+     matrix(id,nrow=5,byrow=TRUE))[c(sample(1:5),sample(6:10))],
+     Block=factor(rep(1:2,each=5)), Column=factor(rep(1:2,each=5)),
+     Row=factor(rep(1:5,2)))
R> prep
```

```
   Trt Block Column Row
1    A     1      1   1
2    B     1      1   2
```



$v = 213$
75×2 replicates
138×1 replicate
$r = 2$
$q_r = 48$
$q_c = 6$
$N = 288$

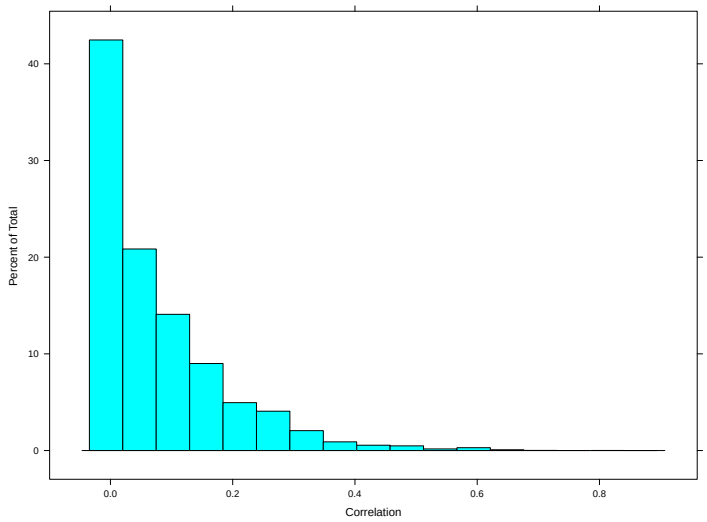
```
  3    C      1       1    3
  4    F      1       1    4
  5    D      1       1    5
  6    E      2       2    1
  7    A      2       2    2
  8    C      2       2    3
  9    G      2       2    4
  10   B      2       2    5
```

```
R> prep.init <- od.init(Column=0.4, Row=0.2, "Column:Row|Column"=0.1,
+     "Column:Row|Row"=0.4)
R> prep.od <- od(fixed= ~Trt, random= ~ Column+Row, rcov= ~ar1(Column):ar1(Row),
+    permute= ~Trt, swap= ~Block, Gstart=prep.init, Rstart=prep.init,
+    maxit=50, data=prep)
R> summary(prep.od)$incidence

  $Column

    1 2
  A 1 1
  B 1 1
  C 1 1
  D 1 0
  E 0 1
  F 1 0
```

Figure 3: Frequency distribution of $\boldsymbol{A} = \{a_{ij}\}, i > j$ on the correlation scale for the 402 individuals in the CBWA pedigree.




```
G 0 1

$Row

  1 2 3 4 5
A 1 0 1 0 0
B 1 0 0 0 1
C 0 0 1 0 1
D 0 1 0 0 0
E 0 0 0 1 0
F 0 0 0 1 0
G 0 1 0 0 0
```

The initial A-value for the starting design was 1.614993, and the optimal A-value at termination was 1.415362 for an approximate 15% gain. The optimal design is binary with respect to *Row* although treatment C is a self-neighbour in the initial random configuration; as noted previously this is not a guaranteed outcome.

The figure above is a real example from the canola breeding program of Canola Breeders WA (CBWA) and we use it to illustrate the use of a numerator relationship matrix ($\boldsymbol{A}$) in design. That is, we assume that $\text{var}(\boldsymbol{u}_g) = \sigma^2_g \boldsymbol{A}$. The 217 entries in the experiment are from a pedigree containing 482 individuals, which was subsequently pruned to 402, that is the 217 trial entries and 185 progenetors. Some indication of the degree of relatedness in the pedigree is shown in Figure 3 where the relative frequencies of the elements in the lower triangle of $\boldsymbol{A}$ on the correlation scale are given.

The $\boldsymbol{A}^{-1}$ matrix is required by the mixed model equations, and like ASReml this is given to `od()` in 3 column sparse form:

```
R> pinv[1:10,]
```

```
    Row Column  Ainverse
1     1      1 0.7897590
2     2      2 2.3440514
3     3      3 0.9543473
4     4      4 1.3209545
5     5      5 0.9997506
6     6      5 0.4995005
7     6      6 0.9997506
8     7      7 1.8990912
9     8      7 0.9990010
10    8      8 2.2989313
```

```
R> str(canola.df)
```

```
'data.frame':        288 obs. of  4 variables:
$ column: Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
$ row   : Factor w/ 48 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
$ block : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
$ geno  : Factor w/ 213 levels "ICPBER-0001",..: 21 83 193 13 137 70 22 47 168 ...
```

The mechanism for including a known relationship matrix in od() parallels that for ASReml, that is, the inverse matrix is given by the *ginv* argument and is linked to the relevent model term with the *ped()* special function. By default, od() will generate a design optimal for those entries in the dataframe.

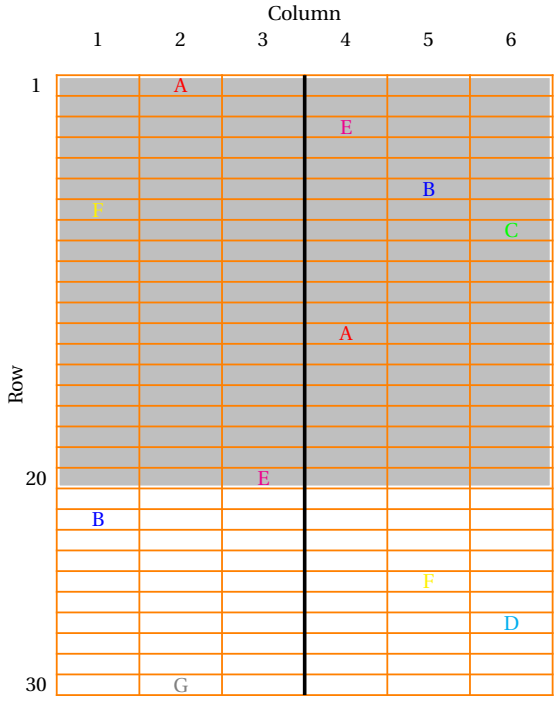
```
R> canola.G <- od.init("ped(geno)" = 1.0, block=0.3, column=0.6, row=0.4 )
R> canola.R <- od.init("column:row|column"=0.2, "column:row|row"=0.5)
R> canola.od <- od(random = ~ ped(geno)+block+column+row,
+     rcov = ~ ar1(column):ar1(row),permute = ~ ped(geno), swap = ~ block,
+     Gstart=Ginit, Rstart=Rinit, ginv=list(geno=pinv), optimize='data',
+     search='anneal',method='PEV',maxit=50, data=canola.df)
```

A term for non-additive genetic effects, specified by the *ide()* special function, has been omitted from the above model, and as a consequence we are assuming (perhaps unrealistically) that the genetic effects are exclusively additive. For this example, there was a gain of approximately 7.0% in the average prediction error variance when the optimal configuration ignoring the known genetic relationship matrix was further optimized using the pedigree information.

7.4 Embedded p-rep designs

The context for this type of design is introduced in Chapter 1 and Smith et al. (2011). Briefly, some quality characteristics of grain samples from field experiments are typically expensive (in economic or throughput terms) to determine. The concept of an *embedded*

Figure 4: An embedded p-rep design.



Full design

$v = 90$
$r = 2$
$q_r = 30$
$q_c = 6$

Shaded area

$v = 90$
30 x 2 + 60 x 1 = 120 plots
$r = 2$
$q_r = 20$
$q_c = 6$

design for this purpose within a larger experiment mitigates the high cost by sampling fewer plots while maintaining statistical rigour.

Given a pre-specified replication scheme, there are several approaches to sampling from the full design to reduce laboratory costs, including:

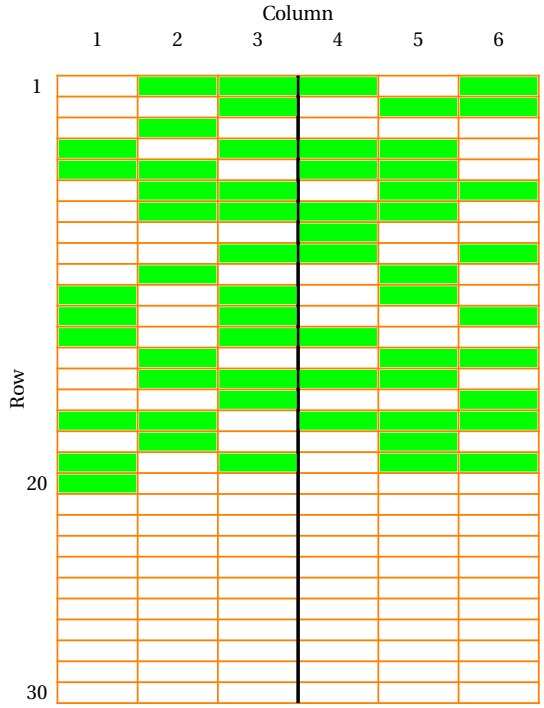
Sample from the full design by choosing one complete replicate and subsetting treatments from a second replicate. This leads to a spatially disjoint scheme.

A conditional approach whereby an optimal *p-rep* design and optimise the full design conditional on fixing the *p-rep* design.

Joinly optimise both areas using a common spatial model and including model term(s) defining the experimental structure. Treatment interchanges that violate the embedded layout are disallowed.

The third option is illustrated here. The full design is a complete block configuration, and the embedded design is a contiguous array of plots in the first two thirds of the area (Figure 4). The two areas are identified by the levels of the factor *Prep*, and the integrity of the embedded design is preserved through the search by declaring the swap boundaries as the intersection of *Prep* and *Replicate*

Figure 5: An example embedded p-rep outcome.



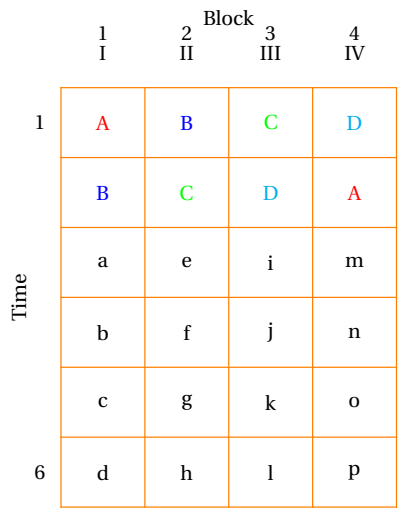

```
R> Ginit <- od.init("~Row"=0.1, "~Column"=0.1, "~Rep"=0.1, "~Prep"=0.1)
R> Rinit <- od.init("~Column:Row|Column"=0.3,"~Column:Row|Row"=0.6)
R> prep.od <- od(fixed = ~ID, random = ~ Prep+Rep+Column+Row,
+     rcov = ~ ar1(Column):ar1(Row), permute = ~ ID, swap = ~ Prep:Rep,
+     Gstart=Ginit, Rstart=Rinit, data=site, maxit=40)
```

An example outcome is shown in Figure 5, where the shaded areas correspond to the replicated genotypes.

7.5 Chain block designs

The chain block design was introduced by Youden & Connor (1953), motivated by applications in the physical sciences where measurements are often made with high precision, and replication can be reduced. They have since proved useful in *sequential* applications where subjects or samples arrive chronologically, possibly from a prior phase process. The real example below is from a two phase process (actually three counting the original field experiment) where flour samples from milled grain were to be scored for viscosity measurements using a rapid viscosity analyzer (RVA). The full experiment is too large to be considered in depth here, so for illustration we initially consider a small synthetic example.

```
R> x <- 1:4
R> cb <- data.frame(Trt = as.vector(rbind(
```



A ... D replicated groups
a ... p unreplicated treatments
$g = 1$
$v = 20$
$k = 6$
$s = 4$
$N = 24$

```
+     matrix(LETTERS[c(x,x%%4+1)],nrow=2,byrow=TRUE),
+     matrix(letters[1:16],nrow=4,byrow=FALSE))), Block = factor(rep(1:4,each=6)),
+     Time = factor(rep(1:6,4)), resolvable = rep(c(rep(TRUE,2),rep(FALSE,4)),4))
R> cb[1:10,]
```

```
   Trt Block Time resolvable
1    A     1    1       TRUE
2    B     1    2       TRUE
3    a     1    3      FALSE
4    b     1    4      FALSE
5    c     1    5      FALSE
6    d     1    6      FALSE
7    B     2    1       TRUE
8    C     2    2       TRUE
9    e     2    3      FALSE
10   f     2    4      FALSE
```

The column labelled *resolvable* is in parallel with the *Trt* factor and is used by od() to modify the behaviour of the *swap* argument. If both swap units have treatments labelled *resolvable=FALSE* then the constraint imposed by *swap* is ignored, and those treatments can move throughout the design. Thus, in the configuration above treatments A-D remain within their assigned blocks, while treatments a-p can be exchanged throughout the design.

```
R> cb.init <- od.init(Block=0.4, Time=0.1, "Block:Time|Time"=0.3)
R> cb.od <- od(fixed= ~Trt, random= ~ Block+Time, rcov=~Block:ar1(Time),
+     permute= ~Trt, swap= ~ Block, Gstart=cb.init, Rstart=cb.init,
+     maxit = 100, data=cb)
R> matrix(cb.od$design$Trt,nrow=6)
```

```
     [,1] [,2] [,3] [,4]
[1,] "j"  "m"  "c"  "a"
[2,] "A"  "B"  "C"  "D"
[3,] "b"  "o"  "k"  "h"
[4,] "l"  "e"  "g"  "f"
[5,] "B"  "C"  "D"  "A"
[6,] "d"  "p"  "i"  "n"
```

The two phase example introduced above has chain block parameters $v = 620$, $k = 16$, $g = (2, 3)$ for $s = 46$ blocks (days), where the last block has ony 15 units. The first phase milling process spanned 62 days, and it was a requirement that the RVA assessment begin as soon as possible after milling commenced, running in parallel with the milling process thus restricting the abiliy to allocate treatments throughout the design.

It is possible that experimental units from the milling phase will carry effects from milling sources of variation into the RVA phase of testing. These potential effects, such as *milling day* or *mill order* must be kept in sync with treatment exchanges throughout the optimization. The pipe symbol "—" can be used in the *permute* formula to specify the factors additional to the treatment factor that are to be permuted. This design took some considerable time to run and the od() call is reproduced below to illustrate the form of the *permute* argument.

```
R> load("rva.RData")
R> rva$MillR <- factor(paste(as.character(rva$MillDay),
+     as.character(rva$MillOrd)))
R> str(rva)

 'data.frame':        735 obs. of  16 variables:
 $ Day            : Factor w/ 46 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Seq            : Factor w/ 16 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 ...
 $ MillRep        : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ MillDay        : Factor w/ 62 levels "1","2","3","4",..: 2 1 1 2 1 1 1 1 1 1 ...
 $ MillOrd        : Factor w/ 10 levels "1","2","3","4",..: 1 3 5 2 6 7 8 4 1 9 ...
 $ Bay            : int  1 3 3 2 1 1 3 1 1 2 ...
 $ Row            : int  6 71 70 29 16 26 79 13 1 53 ...
 $ Col            : int  13 1 3 14 5 13 5 11 2 17 ...
 $ ID             : Factor w/ 508 levels "Alsen","Baxter",..: 342 125 156 276 ...
 $ frep           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ trialUnitID    : int  1517 610 771 1621 879 1537 942 1362 621 1888 ...
 $ compTrialUnitID: int  NA NA NA NA NA 1992 NA NA NA NA ...
 $ repDesc        : Factor w/ 4 levels "Comp","field rep",..: 4 2 4 4 4 1 4 4 4 ...
 $ slDup          : logi  TRUE FALSE FALSE TRUE FALSE FALSE ...
 $ MillR          : Factor w/ 620 levels "1 1","1 10","1 2",..: 111 4
 6 113 7 8 ...

R> rva.G <- od.init(Day=0.9,Seq=0.6, MillRep=0.1,MillDay=0.9,
+     MillOrd=0.4,MillR=1.0)
R> rva.od <- od(fixed = ~ ID, random = ~ Day + Seq + MillRep + MillDay
+     + MillOrd + MillR, permute=~ID|MillRep+MillDay+MillOrd+MillR,
+     swap=~Day, Gstart=rva.G, maxit=50, data=rva)
```

7.6 Partial compositing

The rationale behind partially compositing grain samples as a means of recovering (statistical) information, or indeed in practice as a means of gaining information on genotypes (otherwise lost) with insufficient individual plot samples is detailed elsewhere in these notes. For example, in Figure 4, the 30 replicated genotypes in the shaded area are tested in duplicate while the remaining 60 genotypes are tested as composite samples. That is, a mixed sample from the single plot within the embedded area, and the one outside is prepared for further testing.

As noted earlier, the analysis of these data in ASReml requires special design matrices which can be constructed using the *and()* model function. At the time of writing this model function is not implemted in `od()`, consequently designs that recognise this compositing scheme cannot be constructed strictly in accordance with the appropriate model.

8 Summary

Optimal design with a flexible computing strategy is a powerful tool for design generation, particularly in novel situations, or those with awkward experimental restrictions. On the positive side, the advantages of the technology include:

- Flexibility
  - in terms of the underlying linear model, and
  - in terms of the experimental configuration, given an appropriate computing strategy.
- The model based approach strongly links design and analysis. The model formulae from `od()` provide both an accurate record of the properties of the design, and the means of analysis.
- The computational approach using direct methods promotes this flexibility, and allows computing efficiencies such as exploiting sparsity.

There are, however, unanswered questions and software support issues the include:

- The methodology requires the specification of a linear model, and pre-specified variance parameters. The principal question that arises is that of robustness
  - in terms of the choice of linear mixed model, and
  - sensitivity to variance parameter estimates.
  - Some work on robustness appears in Chan (1999), where it was found that the choice of parameter value within a model class was less important than the choice of variance model family. However, with the range of design models now possible, the issue of robustness warrants further investigation.

- The choice of optimality criterion is not necessarily resolved.
- The construction of an initial design. Software to at least *partially* automate this step is still under construction. It is not possible to anticipate all cases but the ability to generate binary designs for *quasi* regular configurations is a current deficiency.

9 The od() class

`od` *Optimal design*

Description

Generate optimal categorical designs under a general linear mixed model.

Usage

```
od(fixed = ~1, random = ~NULL, rcov = ~NULL, permute = ~NULL,
   swap = ~NULL, ginv = list(), Gstart = list(), Rstart = list(),
   data = sys.parent(), search = c("anneal","tabu", "random"),
   method = c("avalue","PEV"), maxit = 1, trace = FALSE, debug = FALSE)
```

Arguments

`fixed`	a formula specifying the fixed model terms.
`random`	a formula specifying the random model terms.
`rcov`	a model formula specifying the error structure.
`permute`	a formula containing a single term specifying the factor in the `fixed` or `random` set that is to be permuted. The `"|"` operator can be used to associate this factor with other terms in the model that should be permuted in parallel.
`swap`	a formula nominating which factor(s) define legal treatment exchanges during the optimisation process. Rows of `W = cbind(X,Z)` are only permuted within levels of the swap factor (or the intersection of levels if `swap` is of the form `'~ A:B'`).
`ginv`	a list object associating a known inverse treatment relationship matrix with a random term in the model. The inverse is given in sparse form `(row, column, value)` as lower triangle row-wise.
`Gstart`	a list object containing initial values for the random components, typically generated from a call to `od.init`.
`Rstart`	a list object containing initial values for the residual components, typically generated from a call to `od.init`.
`data`	a dataframe in which to resolve the terms in the model formulae.

`search` a character string specifying the search strategy. The default, `"anneal"`, specifies classical simulated annealing, `"tabu"` initiates a robust tabu search (Taillard, 1991) while `"random"` simply exchanges treatments in an undirected manner.

`method` the optimality criterion. `"avalue"` is typically used when treatment effects are considered as fixed, and optimizes the average pairwise variance of elementary contrasts. `"PEV"` may be more appropriate for random effects, particularly when a known treatment covariance matrix is present, and optimizes the trace of the inverse coefficient matrix for treatments.

`maxit` the number of *cooling* loops for simulated annealing, or tabu loops if the search method is `tabu`, otherwise the number of random exchanges.

`trace` if `TRUE` then the optimization progress is reported. Currently `FALSE` does nothing.

`debug` if `TRUE` various R structures are returned for debug purposes. If numeric (1 or 2), intermediate structures and computations are printed from the numerical functions.

Details

The mixed model equations are formed from the model given by the `fixed`, `random` and `rcov` formulae, and the variance parameters pre-specified in `Gstart` and `Rstart`. The objective function is calculated for the model (treatment) term given in the `permute` formula. The objective function (`avalue` or `PEV`) is optimized under the supervision of the `search` strategy for successive exchanges of pairs of treatments, subject to the conditions set by the terms in `swap`, until a stopping condition is satisfied. Optimization proceeds until `maxit` is exceeded (`anneal`, `tabu` and `random`), or the temperature change is lower than the specified minimum for simulated annealing (see `od.options`).

The cooling schedule for simulated annealing can be either `geometric` or `lundy`, and is set along with other annealing options in `od.options`.

Special functions may be used in model formulae; currently the only recognised functions are `id, ar1, ped` and `ide`.

Value

A list object of class `od` with the following components:

`call` the `od()` function call.

`design` the dataframe with the factor nominated in the `permute` formula in design order.

`criterion` the final value of the optimality criterion.

`permutation`	the permutation, or design order, of the rows of `data`.
`Gstart`	initial values
`Rstart`	initial values

`od.init` *initial values for od()*

Description

creates a list object containing initial values for function `od`.

Usage

```
od.init(...)
```

Arguments

`...`	a series of comma separated `name = value` pairs, where `name` is a character string matching a term in an `od` model, and `value` is the associated numeric starting value. If `name` contains special characters then it must be quoted. The `"|"` operator identifies individual components of direct product structures.

Value

A list object named by the arguments containing starting values for `od`.

`od.options` *Settings for od()*

Description

Allows various options that affect the behaviour of `od` to be set or examined.

Usage

```
od.options(...)
```

Arguments

`...`	a series of comma separated `name = value` pairs, where `name` is a character string matching one of the options below, and `value` is the allowable setting.

Details

If called with no arguments then a list of current option values is returned. The list of options is held in an environment `.ODenv` in the `od` database.

Value

A list of current options and their values if called as `od.options()`, otherwise `invisible()`.

Options used in OD

cool the cooling schedule for simulated annealing, recognised values are `"lundy"` (default), or `"geometric"`.

cycles if search is `anneal`, this is the number of random interchanges to evaluate in exploring the neighbourhood of a solution within a temperature cycle, default=1000.

alpha temperature reduction coefficient used in `geometric` cooling, default=0.7.

beta temperature reduction coefficient used in `lundy` cooling, default=0.7.

prob transition probability, default=0.95.

tstop temperature at which to terminate the search (simulated annealing), default = 1e-5. Optimization terminates when the temperature falls below `tstop`, or `maxit` is exceeded.

Examples

```
## Get all options
od.options()

## Set the simulated annealing cooling schedule
od.options(cool="geometric")
```

Bibliography

Atkinson, A. C. (1982). Developments in the design of experiments. *International Statistical Review* 50 161–177.

Atkinson, A. C. & Donev, A. N. (1992). *Optimum Experimental Designs*. Oxford: Clarendon Press.

Atkinson, A. C., Donev, A. N. & Tobias, R. D. (2007). *Optimum Experimental Designs, with SAS*. Oxford University Press.

Battiti, R. & Tecchiolli, G. (1994). The reactive tabu search. *ORSA Journal on Computing* 6 126–140.

Bueno Filho, J. S. S. & Gilmour, S. G. (2003). Planning incomplete block experiments when treatments are genetically related. *Biometrics* 59 375–381.

Butler, D. G., Eccleston, J. A. & Cullis, B. R. (2008). On an approximate optimality criterion for the design of field experiments under spatial dependence. *Australian and New Zealand Journal of Statistics* 50 295–307.

Butler, D. G., Smith, A. B. & Cullis, B. R. (2012). On the design of experiments where treatments are correlated. *Biometrics* submitted.

Chan, B. S. P. (1999). *The design of field experiments when the data are spatially correlated.* Phd thesis, Department of Mathematics, University of Queensland.

Coombes, N. (2002). *The reactive tabu search for efficient correlated experimental designs*. Phd thesis, Liverpool John Moores University.

Cox, D. R. & Reid, N. (2000). *The Theory of the Design of Experiments*. Boca Raton: Chapman and Hall.

Cullis, B. R. & Gleeson, A. C. (1991). Spatial analysis of field experiments - an extension to two dimensions. *Biometrics* 47 1449–1460.

Cullis, B. R., Gogel, B. J., Verbyla, A. P. & Thompson, R. (1998). Spatial analysis of multi-environment early generation variety trials. *Journal of the Royal Statistical Society, Series B* 39 1–38.

Cullis, B. R., Smith, A. B. & Coombes, N. E. (2006a). On the design of early generation variety trials with correlated data. *Journal of Agricultural, Biological, and Environmental Statistics* 11 381–393.

Cullis, B. R., Smith, A. B. & Coombes, N. E. (2006b). On the design of early generation variety trials with correlated data. *Journal of Agricultural, Biological and Environmental Statistics* 11 381–393.

Eccleston, J. A. & Whitaker, D. (1999). On the design of optimal change-over experiments through multi-objective simulated annealing. *Statistics and Computing* 9 37–42.

Elliot, L. J., Eccleston, J. A. & Martin, R. J. (1999). An algorithm for the design of factorial experiments when the data are correlated. *Statistics and Computing* 9 195–201.

Federer, W. T. & Raghavarao, D. (1975a). On augmented designs. *Biometrics* 31 29–35.

Fisher, R. A. (1935). *The Design of Experiments*. Edinburgh: Oliver and Boyd.

Gilmour, A. R., Cullis, B. R. & Verbyla, A. P. (1997). Accounting for natural and extraneous variation in the analysis of field experiments. *Journal of Agricultural, Biological, and Environmental Statistics* 2 269–273.

Glover, F. (1989). Tabu search - Part I. *ORSA Journal on Computing* 1 190–206.

Glover, F. (1990). Tabu search - Part II. *ORSA Journal on Computing* 2 4–32.

Goos, P. (2002). The optimal design of blocked and split-plot experiments. In P. Bickel, P. Diggle, S. Fienberg, K. Krickeberg, I. Olkin, N. Wermuth & S. Zeger, eds., *Lecture Notes in Statistics*, 164. Springer.

Henderson, C. R. (1976). A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. *Biometrics* 32 69–83.

Kiefer, J. (1959). Optimum experimental designs. *Journal of the Royal Statistical Society, Series B* 21 272–319.

Kiefer, J. (1974). General equivalence theory for optimum designs (approximate theory). *Annals of Statistics* 2 849–879.

Kiefer, J. & Wolfowitz, J. (1960). The equivalence of two extremum problems. *Canadian Journal of Mathematics* 12 363–366.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimisation by simulated annealing. *Science* 220 671–680.

Lin, C. S. & Poushinsky, G. (1983). A modified augmented design for an early stage of plant selection involving a large number of test lines without replication. *Biometrics* 39 553–561.

Lundy, M. (1985). Applications of the annealing algorithm to combinatorial problems in statistics. *Biometrika* 72 191–198.

Martin, R. J. (1986). On the design of experiments under spatial correlation. *Biometrika* 73 247–277.

Martin, R. J. & Eccleston, J. A. (1997). Construction of optimal and near optimal designs for dependent observations using simulated annealing. Research Report 479/97, Department of Probability and Statistics.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). Equations of state calculations by fast computing machine. *Journal of Chemical Physics* 21 1087–1091.

Mrva, K. & Mares, D. J. (2001). Induction of late maturity $\alpha-$amylase in wheat by cool temperature. *Australian Journal of Agricultural Research* 52 477–484.

Osman, I. H. (1991). *Metastrategy simulated annealing and tabu search algorithms for combinatorial optimisation problems*. Phd thesis, University of London, Imperial College, UK.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1996). *Numerical recipes in C*. Cambridge, UK: Cambridge University Press.

Raghavarao, D. (1971). *Constructions and combinatorial problems in design of experiments*. New York: John Wiley and Sons.

Robinson, G. K. (1991). That blup is a good thing: The estimation of random effects. *Statistical Science* 6 15–51.

Silvey, S. D. (1980). *Optimal Design*. London: Chapman and Hall.

Smith, A. B., Cullis, B. R., Gilmour, A. R. & Thompson, R. (1998). Multiplicative models for interaction in spatial mixed model analyses of multi-environment trial data. In *Proceedings of the International Biometrics Conference*. Capetown.

Smith, A. B., Thompson, R., Butler, D. G. & Cullis, B. R. (2011). The analysis of variety trials using mixtures of composite and individual plot samples. *Journal of the Royal Statistical Society, Series C* 60 437–455.

Williams, E. R. & John, J. A. (1996). Row-column factorial designs for use in agricultural field trials. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 45 39–46.

Williams, E. R., John, J. A. & Whitaker, D. R. (2006). Construction of resolvable spatial row-column designs. *Biometrics* 62 103–108.

Woodruff, D. L. & Zemel, E. (1993). Hashing vectors for tabu search. *Annals of Operations Research* 41 123–137.

Wu, C. F. (1978). Some algorithmic aspects of the theory of optimal designs. *Annals of Statistics* 6 1286–1301.

Yates, F. (1936a). A new method of arranging variety trials involving a large number of varieties. *Journal of Agricultural Science (Cambridge)* 26 424–455.

Youden, W. J. & Connor, W. S. (1953). The chain block design. *Biometrics* 9 127–140.

Chapter

J

Whole Genome Analysis using Linear Mixed Models

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, 2010). 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 workshop 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 is available 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 achieved computationally using the R package **ASReml-R**. The simulation results given in Section 3 and 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 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 further use in **wgaim**. The converted objects retain a similar structure to objects 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, double 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 of **wgaim**.

Throughout the WGAIM procedure the underlying linear mixed model analysis is achieved using the highly flexible R software package **ASReml-R**, built as a front end

wrapper for the more sophisticated 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 theoretical method

The WGAIM approach is a forward selection method that uses a whole genome approach to genetic analysis at each step. 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

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_e\boldsymbol{u}_e + \boldsymbol{Z}_g\boldsymbol{g} + \boldsymbol{e}, \tag{1}$$

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 (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

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

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

$$\begin{aligned} \boldsymbol{g} &= \sum_{k=1}^{c} \sum_{j=1}^{m_k - 1} (\boldsymbol{m}_{k:j}\lambda_{k:j,j} + \boldsymbol{m}_{k:j,j+1}\lambda_{k:j+1,j})a_{k:j} + \boldsymbol{p} \\ &= \boldsymbol{M}\boldsymbol{\Lambda}\boldsymbol{a} + \boldsymbol{p}, \end{aligned} \tag{2}$$

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

$$\begin{aligned} \boldsymbol{g} &= \sum_{k=1}^{c} \sum_{j=1}^{m_k - 1} (\boldsymbol{m}_{k:j} + \boldsymbol{m}_{k:j,j+1})\lambda_{k:j}a_{k:j} + \boldsymbol{p} \\ &= \boldsymbol{M}\boldsymbol{\Lambda}_E\boldsymbol{a} + \boldsymbol{p}, \end{aligned} \tag{3}$$

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

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_e\boldsymbol{u}_e + \boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a} + \boldsymbol{Z}_g\boldsymbol{p} + \boldsymbol{e}. \tag{4}$$

After the fitting of (4) 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 (4) 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

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_e\boldsymbol{u}_e + \boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a} + \boldsymbol{Z}_{g,k}\boldsymbol{M}_{E,k}\boldsymbol{\delta}_k + \boldsymbol{Z}_g\boldsymbol{p} + \boldsymbol{e}, \tag{5}$$

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

$$U_k(0) = -\frac{1}{2}\left(\mathrm{tr}(\boldsymbol{C}_{k,k}) - \frac{1}{\sigma^2\gamma_a^2}\tilde{\boldsymbol{a}}_k^T\tilde{\boldsymbol{a}}_k\right), \tag{6}$$

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 (6) 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

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

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

$$t_{k:j}^2 = \frac{\tilde{a}_{k:j}^2}{\text{var}(\tilde{a}_{k:j})}. \tag{7}$$

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 (5) 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 (4). 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 (4) 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

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}_{a,s}\boldsymbol{M}_{E,s}\boldsymbol{a}_s + \boldsymbol{Z}_e\boldsymbol{u}_e + \boldsymbol{Z}_g\boldsymbol{p} + \boldsymbol{e}, \tag{8}$$

where $\boldsymbol{Z}_{a,s}$ and $\boldsymbol{M}_{E,s}$ contain the the appropriate columns of $\boldsymbol{Z}_g$ and $\boldsymbol{M}_E$ for the selected QTL. This complete approach is known as the WGAIM algorithm.

2.1 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) 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) can be written as an approximate marker QTL regression model

$$\boldsymbol{g} = \boldsymbol{M}\boldsymbol{a}_M + \boldsymbol{p} \tag{9}$$

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 (9) with independent marker QTL effects will be less efficient than the interval mapping equivalent (2).

2.2 Recent advances: WGAIM v1.0+

WGAIM is always being developed to improve its efficiency and stability as well as advance its capabilities. Recent research in Verbyla & Taylor (2011) has shown the WGAIM method can be improved through a slight alteration is the selection process as well as using a random effects formulation for the selected QTL. Verbyla & Taylor (2011) also show that its possible to efficiently include high dimensional genetic components in the linear mixed model formulation. These points are discussed below.

2.2.1 The outlier statistics

There is a short relevant point in Verbyla & Taylor (2011) 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 $t_k^2 = t_{k:1}^2$.

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. It was discovered after many simulations that a better choice would be to only use the interval outlier statistic to guide the selection process.

2.2.2 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 $\boldsymbol{a}_s \sim N(\boldsymbol{0}, \sigma_{as}^2 \boldsymbol{I})$ so the selected QTL effects are assumed to be random and have a different variance to the unselected effects. This makes sense as selected QTL effects exhibit variation from zero because they are QTL. Thus random version of WGAIM mimics a mixture distribution, in which QTL come from one distribution and non-QTL come from another distribution. The two distributions differ in their variances and not their means.

2.2.3 High dimensional analysis

A major issue concerns the possible high-dimension of M_λ in (4) at any stage of the selection. For ease of presentation we consider the first step of selection. The matrix M_λ is of size $r \times m - c$, and $m - c$ may be very large, larger than r. If $r > m - c$ there is no need to invoke dimension reduction. If $r < m - c$, it is possible to carry out efficient computations not in dimension $m - c$ but in dimension r which may be much smaller, and subsequently to recover the $m - c$ effects necessary for selection.

For generality, suppose that$\boldsymbol{a} \sim N(\boldsymbol{0}, \sigma_a^2 \boldsymbol{G}_a)$. In WGAIM, $\boldsymbol{G}_a = \boldsymbol{I}_{m-c}$. Taylor & Verbyla (2011b) consider an alternative approach based on penalized likelihoods and in that case at stage k of the iterative process involved, $\boldsymbol{G}_a = \boldsymbol{W}_k$ where $\boldsymbol{W}_k$ is a diagonal matrix of known weights. In these cases, $\boldsymbol{G}_a$ is a known matrix and the results of this section will hold for any method in which this is the case or can be constructed to be so. Note that while WGAIM involved pseudo-markers for intervals, these can be replaced by marker scores.

Thus suppose for the moment that no QTL have been selected ($s = 0$ in the discussion above). The variance model for the random effects $\boldsymbol{M}_E\boldsymbol{a}$ is of the form

$$\sigma_a^2 \boldsymbol{M}_E \boldsymbol{G}_a \boldsymbol{M}_E^T \tag{10}$$

which is of size $r \times r$. The matrix $\boldsymbol{M}_E\boldsymbol{G}_a\boldsymbol{M}_E^T$ is known. Then we can re-formulate the working mixed model (4) as follows. Let

$$\boldsymbol{Z}_\lambda = (\boldsymbol{M}_E \boldsymbol{G}_a \boldsymbol{M}_E^T)^{1/2}$$

denote the matrix square root; this can be found using the singular value decomposition (Golub & van Loan, 1996). If we define an $r \times 1$ vector $\boldsymbol{a}^*$ as

$$\boldsymbol{a}^* \sim N(\boldsymbol{0}, \sigma_a^2 \boldsymbol{I}_r)$$

the working model term for interval effects $\boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a}$ can be replaced

$$\boldsymbol{Z}_g \boldsymbol{Z}_\lambda \boldsymbol{a}^*$$

and the same variance model (10) is obtained. Thus we can estimate r effects rather than $m - c$. Most importantly for QTL analysis however, the $m - c$ effects can be recovered. Note firstly that the BLUP for $\boldsymbol{a}^*$ is given by

$$\tilde{\boldsymbol{a}}^* = \sigma_a^2 (\boldsymbol{M}_E \boldsymbol{G}_a \boldsymbol{M}_E^T)^{1/2} \boldsymbol{Z}_g^T \boldsymbol{P} \boldsymbol{y}$$

The BLUP of $\boldsymbol{a}$ is given by

$$\tilde{\boldsymbol{a}} = \sigma_a^2 \boldsymbol{G}_a \boldsymbol{M}_E^T \boldsymbol{Z}_g^T \boldsymbol{P} \boldsymbol{y}$$

so that we can write

$$\tilde{\boldsymbol{a}} = \boldsymbol{G}_a \boldsymbol{M}_E^T (\boldsymbol{M}_E \boldsymbol{G}_a \boldsymbol{M}_E^T)^{-1/2} \tilde{\boldsymbol{a}}^* \tag{11}$$

thereby recovering the estimated size of potential QTL in each interval.

To calculate the outlier statistic given by (7), we need the variance of the best linear unbiased predictor (BLUP) of $\boldsymbol{a}$. If $\tilde{\boldsymbol{a}}$ denotes the BLUP of $\boldsymbol{a}$, the variance matrix is

$$\text{var}(\tilde{\boldsymbol{a}}) = \boldsymbol{G}_a \boldsymbol{M}_E^T (\boldsymbol{M}_E \boldsymbol{G}_a \boldsymbol{M}_E^T)^{-1/2} (\tilde{\boldsymbol{a}}^*) (\boldsymbol{M}_E \boldsymbol{G}_a \boldsymbol{M}_E^T)^{-1/2} \boldsymbol{M}_{\boldsymbol{E}} \boldsymbol{G}_a \tag{12}$$

and so the variance matrix of the BLUPs of $\boldsymbol{a}^*$ is required, again a lower dimensional calculation after fitting the reduced dimensional model. The diagonal elements of $(\tilde{\boldsymbol{a}})$ can be used together with (11) to calculate the outlier statistics (7).

2.2.4 Links to genomic prediction

There are important consequences for genomic prediction and selection (Meuwissen et al., 2001) in the development of the high dimensional theory derived above.

Note, the BLUP for $\boldsymbol{a}^*$ can be rewritten as

$$\tilde{\boldsymbol{a}}^* = \boldsymbol{M}_E \tilde{\boldsymbol{a}}$$

so that for genomic prediction, we need only form $\tilde{\boldsymbol{a}}^*$. The individual BLUPs $\tilde{\boldsymbol{a}}$ obtained through back transformation are available but not required.

2.3 Final assessment of significance of QTL

Once all the QTL are selected they are summarized differently depending on the formulation.

2.3.1 Fixed effects formulation

The summary of the selected QTL in this formulation can proceed using a Wald statistic. Let $\hat{a}_{jk}$ be the fixed effect estimate of the QTL a_{jk} with variance $var(\hat{a}_{jk}) = \sigma^2_{PEV,jk}$. The wald statistic for this QTL is then given by

$$z_{jk} = \frac{\tilde{a}_{jk}}{\sigma_{PEV,jk}}$$

Assuming $Z \sim N(0,1)$ then a p-value is calculated in the usual manner using

$$\Pr\big(Z < -\frac{\tilde{a}_{jk}}{\sigma_{PEV,jk}} | \hat{a}_{jk} < 0\big)$$

and

$$1 - \Pr\big(Z < -\frac{\tilde{a}_{jk}}{\sigma_{PEV,jk}} | \hat{a}_{jk} > 0\big)$$

2.3.2 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_{jk} say, given the data is used. That is under the normality assumptions for a linear mixed model,

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

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_{jk}, that is the estimated size of the QTL $\tilde{a}_{jk}$, and the variance is the prediction error variance (PEV) of a_{jk}. Thus the proper assessment of the impact of the QTL involves determining how far the distribution is from zero. Clearly, distributions that are far from zero indicate a strong QTL effect whereas distributions which include zero will indicate a weaker QTL. This can be quantified by calculating a probability somewhat like a p-value, but for which values close to 1 indicate the QTL is strong. Thus for estimated effects that are positive, the probability that the true (random) QTL effect is greater than zero is calculated, while for estimated effects that are negative the probability that the true (random) QTL effect is less than zero is calculated.

Thus,

$$\Pr(a_{jk} < 0 | \boldsymbol{y}_2) = \Pr\left(Z < -\frac{\tilde{a}_{jk}}{\sigma_{PEV,jk}}\right)$$

and

$$\Pr(a_{jk} > 0|\boldsymbol{y}_2) = 1 - \Pr\left(Z < -\frac{\tilde{a}_{jk}}{\sigma_{PEV,jk}}\right)$$

These probabilities can be found approximately by substituting the estimated $\sigma_{PEV,jk}$ from the analysis. While LOD scores are not appropriate for WGAIM, the statistic z_{jk} can be used to calculate a LOD score if this is deemed necessary.

2.4 Current research

Currently only QTL analysis of univariate traits are possible with the accompanying software, **wgaim**. However, several extension are being developed and itemized here.

- A multivariate version of the WGAIM algorithm has been researched and preliminary papers have been submitted (Verbyla & Cullis, 2010; Verbyla et al., 2010). The software implementation of this multivariate approach is currently being tested.
- An implementation of WGAIM for populations with multiple parents is also being researched. This has shown to be useful in determining QTL for emerging complex breeding populations such as the Multi-parent Advanced Genetic Inter-Cross wheat population being developed in CSIRO, Plant Industry.
- Lastly, research is also currently being conducted to determine the inclusion of higher order effects such as epistatic interactions into the WGAIM approach.

We are hopeful that all these new approaches will be implemented in future releases of **wgaim**.

2.5 Summary

To summarise the WGAIM algorithm provides an efficient method for whole genome interval or marker QTL analysis. As the algorithm is developed through the extension of a flexible linear mixed model it provides an appropriate mechanism for more difficult analysis of experiments that usually occur in plant or animal breeding trials.

3 Simulations

In this section the performance of the fixed and random WGAIM methodologies is examined through simulation studies. Much of the information contained here can be found in more detail in Verbyla et al. (2007) and Verbyla & Taylor (2011). The simulation studies were constructed in a similar manner to the multiple chromosome QTL simulation studies in Broman & Speed (2002).

3.1 Details

The number of genetic lines and hence the population size considered is $l = 100, 200$ and 400. These values were chosen to reflect the size of population common in crop trials. In all simulations, 2 replicates of each line were generated. This differs from Broman & Speed (2002) who consider a single replicate for each line. However, our simulations are constructed to ensure that our results are comparable to those of Broman & Speed (2002).

The simple model for the simulated $y_{ij}, i = 1, \ldots, l; j = 1, 2$ is given by

$$y_{ij} = \mu + g_i + e_{ij}$$

where $e_{ij} \sim N(0, \sigma^2)$ are independent random variables. The genetic line effects g_i are given by

$$g_i = \sum_{s=1}^{S} q_{is} a_s + p_i$$

where there are S QTL, and q_{is} for QTL s is either -1 or 1 depending on the parental allele that line i carries. The polygenic effects $p_i \sim N(0, \sigma_g^2)$ are independent and also independent of e_{ij}. In all simulations, $\sigma^2 = 1$, $\sigma_g^2 = 0.5$ and $a_s = 0.378$.

Genetic lines are assumed to have a total of 9 chromosomes. The underlying chromosomal structures are 100cM in length, with 11 equally spaced markers. Broman & Speed (2002) locate the QTL at markers, but in our study, the QTL are located at midpoints of intervals in configurations to be discussed below. This is the more likely situation in practice. The genetic lines are simulated on the basis of this underlying structure, and hence each line in the simulation has scores on the 11 markers for each chromosome and on the QTL.

For each population size ($l = 100, 200, 400$), a linkage map was constructed on the basis of the simulated marker scores, and also including the simulated QTL scores to accurately place the QTL on the estimated linkage map. This was required to confirm the detection or non-detection of a QTL. Note that while the underlying structure has chromosomes of length 100cM and equally spaced markers, the estimated linkage map for a chromosome varied in length from 80 to 120cM, with marker spacing varying from 4cM to 25cM. The **qtl** package in R was used to develop the linkage map and then to check the order of the markers and QTL. The final linkage map was used for all simulations for a particular population size. Thus map uncertainty was included in the simulation study.

Table 1: Rate of correct identification of each QTL and the total mean number out of 7.

		Interval							
Size	Method	C1.4	C1.8	C2.4	C2.8	C3.6	C4.4	C5.1	Total
100	WGAIM	0.470	0.445	0.468	0.722	0.679	0.603	0.564	3.951
	CIM7	0.132	0.056	0.208	0.223	0.335	0.241	0.181	1.376
200	WGAIM	0.940	0.952	0.623	0.733	0.790	0.787	0.773	5.598
	CIM7	0.706	0.810	0.234	0.330	0.584	0.555	0.614	3.833
400	WGAIM	0.961	0.968	0.937	0.918	0.980	0.994	0.997	6.755
	CIM7	0.900	0.946	0.919	0.896	0.990	0.989	0.998	6.638

In this simulation $S = 7$ QTL are strategically placed on the linkage map. The locations of the QTL are denoted by Ck.j where Ck is chromosome k and j is interval j on that chromosome. QTL are located at the midpoints of the intervals C1.4, C1.8, C2.4, C2.8, C3.6, C4.4, C5.1, unlike Broman & Speed (2002) who assumed that each QTL was located at the left-hand marker of each interval. The favorable QTL allele is coded as 1 for all but the second QTL C1.8, for which it is -1. Thus the two QTL on chromosome 1 are in repulsion while the two QTL on chromosome 2 are in coupling. The proportion of the total line mean variance due to the 7 QTL was $7\times0.378^2/(7\times0.378^2+1) = 1/(1+1) = 0.50$, again matching Broman & Speed (2002)). There are additional terms in this calculation for the QTL in coupling and repulsion due to linkage, but these additional terms cancel.

Table 2: Mean number of linked extraneous QTL on chromosomes 1 to 5 and unlinked extraneous QTL on chromosomes 6 to 9.

		Linked						Unlinked
Size	Method	C1	C2	C3	C4	C5	Total	C6-C9
100	WGAIM	0.227	0.203	0.150	0.100	0.069	0.749	0.199
	CIM7	0.038	0.092	0.040	0.023	0.005	0.198	0.005
200	WGAIM	0.098	0.145	0.105	0.144	0.049	0.541	0.090
	CIM7	0.043	0.258	0.044	0.105	0.018	0.468	0.001
400	WGAIM	0.032	0.029	0.029	0.009	0.008	0.107	0.025
	CIM7	0.038	0.025	0.009	0.009	0.001	0.082	0.003

3.2 Fixed WGAIM vs CIM

In this section the methods of QTL analysis chosen for comparison are fixed WGAIM, and CIM with 7 co-factors (CIM7), the latter chosen by forward selection. Broman & Speed (2002) found that CIM7 performed well in their simulation study, and as the correct number of cofactors are included our comparison is again with the best CIM approach. Direct comparison between WGAIM and CIM is difficult and thorough details of how this

Table 3: Rate of identification of the QTL in coupling and repulsion. Identification is presented by two by two tables, with, for example, C2.4 on the left and C2.8 on the top of each two by two table. D denotes correctly detected or identified and $\bar{D}$ means not identified.

		Repulsion				Coupling			
		WGAIM		CIM7		WGAIM		CIM7	
Pop. size		$\bar{D}$	D	$\bar{D}$	D	$\bar{D}$	D	$\bar{D}$	D
100	$\bar{D}$	0.418	0.112	0.830	0.038	0.057	0.475	0.574	0.218
	D	0.137	0.333	0.114	0.018	0.221	0.247	0.203	0.005
200	$\bar{D}$	0.017	0.043	0.095	0.199	0.060	0.317	0.468	0.298
	D	0.031	0.909	0.095	0.611	0.207	0.416	0.202	0.032
400	$\bar{D}$	0.015	0.024	0.018	0.082	0.006	0.057	0.007	0.074
	D	0.017	0.944	0.036	0.864	0.076	0.861	0.097	0.822

is achieved for this simulation can be found in Verbyla et al. (2007). For each population size 2000 simulations were carried out.

The correct detection rates for individual QTL are presented in Table 1. CIM7 performs quite badly for population size 100. This is particularly true for the QTL in coupling. The performance of both methods improves as the population size increases. WGAIM again has the best overall detection rates.

There are 7 QTL in the simulations. The total number of correctly identified QTL for the 3 methods and 3 population sizes are also given in Table 1. The results found in our study with regard to CIM are consistent with those found by Broman & Speed (2002). All three methods improve their power as the population size increases. The mean number of correctly identified QTL using WGAIM is always higher than for CIM7, often considerably so. This shows that WGAIM is much more powerful than CIM at finding true QTL.

Table 4: Proportion of the 200 simulations in which the QTL was detected for WGAIM and random WGAIM (RWGAIM) analyses.

Population	Method	C1.4	C1.8	C2.4	C2.8	C3.6	C4.4	C5.1	Total
100	WGAIM	0.350	0.465	0.675	0.425	0.700	0.590	0.695	3.900
	RWGAIM	0.355	0.455	0.685	0.450	0.715	0.615	0.710	3.985
200	WGAIM	0.940	0.915	0.650	0.595	0.760	0.825	0.800	5.485
	RWGAIM	0.945	0.905	0.685	0.615	0.770	0.820	0.810	5.550
400	WGAIM	0.955	0.935	0.900	0.900	0.985	1.000	0.995	6.670
	RWGAIM	0.955	0.940	0.915	0.915	0.985	1.000	0.995	6.705

The rates of detection of extraneous QTL are presented in Table 2. The mean numbers of linked extraneous QTL are presented for each chromosome, together with an overall total. The mean number of unlinked extraneous QTL are presented as a total for chro-

Table 5: Two way tables for the QTL in repulsion (C1.4 and C1.8) with proportions of the 200 simulations for each population size for the combinations of non-detected $\bar{D}$ and detected D QTL. C1.4 is on the left and C1.8 on the top of each 2×2 table.

			Population size					
			100		200		400	
Type	Method		$\bar{D}$	D	$\bar{D}$	D	$\bar{D}$	D
Coupling	WGAIM	$\bar{D}$	0.445	0.205	0.025	0.035	0.040	0.005
		D	0.090	0.260	0.060	0.880	0.025	0.930
	RWGAIM	$\bar{D}$	0.440	0.205	0.030	0.030	0.040	0.005
		D	0.105	0.250	0.070	0.875	0.020	0.935
Repulsion	WGAIM	$\bar{D}$	0.085	0.240	0.055	0.295	0.025	0.075
		D	0.490	0.185	0.350	0.300	0.075	0.825
	RWGAIM	$\bar{D}$	0.075	0.240	0.055	0.260	0.020	0.065
		D	0.475	0.210	0.330	0.355	0.065	0.850

mosomes 6 to 9. The rates for linked extraneous QTL show that for population size 100, where correct detection is difficult, WGAIM has higher rates than CIM7. For $l = 200$, WGAIM has higher rates, except for chromosome 2 where the two QTL are in coupling. Here CIM7 has much higher rates. The total rates are very similar for both methods. For a population size $l = 400$, the rates for both methods are very good and are very similar. The detection of unlinked extraneous QTL is always higher for WGAIM, but decreases as the population size increases. Thus the increased power of detecting true QTL that WGAIM affords, is accompanied by an increase (albeit small) in detection of false QTL.

Table 3 gives the rates for correct identification of QTL in repulsion and coupling presented as two-way tables. For the two QTL in repulsion, CIM7 is very poor for population sizes of 100 and 200. There is considerable improvement for a populations size of 400. In contrast, WGAIM gradually improves detection of both QTL, and has much better rates of detection than CIM across the board, with a marked improvement in performance with a population size of 200. The patterns are similar for two QTL in coupling. Again CIM7 performs poorly for population sizes of 100 and 200, improving considerably for a size of 400. WGAIM gradually improves detection of both QTL, and has much better rates of detection than CIM7 across the board. This study highlights the difficulty in detecting QTL in coupling for small and moderate populations sizes, even with a more powerful method, namely WGAIM.

3.3 Random WGAIM vs Fixed WGAIM

In this section the original fixed WGAIM method is compared to the newer random WGAIM (RWGAIM) method. For each population size 200 simulations were carried out.

Table 4 presents the proportion of simulations in which each QTL was successfully found using standard WGAIM and RWGAIM. The results are very similar with RWGAIM

Table 6: Proportion of the 200 simulations in which false QTL were detected for the 4 methods of analysis. Both linked (selected QTL are on chromosomes with QTL) and unlinked (selected QTL are on chromosomes without QTL) are presented.

	Population size					
	100		200		400	
Method	Linked	Unlinked	Linked	Unlinked	Linked	Unlinked
WGAIM	0.770	0.275	0.445	0.125	0.115	0.030
RWGAIM	0.675	0.255	0.390	0.105	0.105	0.025

finding marginally more QTL for all population sizes. The effect of using random QTL effects is minimal on the good performance of the general WGAIM approach.

Two-way tables for the proportions of QTL detected for the two QTL in coupling and repulsion are given in Table 5. Results for WGAIM and RWGAIM are very similar. Although the detection rates compared to repulsion were lower, RWGAIM shows some improvement over WGAIM in detecting both QTL for all population sizes.

Table 7: Mean estimated size of QTL effects with empirical standard deviations for each method across the 200 simulations for each population size.

		Population Size		
Method	Interval	100	200	400
WGAIM	C1.4	0.503 (0.127)	0.430 (0.131)	0.391 (0.073)
	C1.8	-0.478 (0.110)	-0.439 (0.144)	-0.377 (0.109)
	C2.4	0.576 (0.146)	0.471 (0.144)	0.381 (0.085)
	C2.8	0.585 (0.188)	0.466 (0.112)	0.390 (0.083)
	C3.6	0.501 (0.144)	0.382 (0.110)	0.395 (0.051)
	C4.4	0.480 (0.122)	0.384 (0.108)	0.369 (0.047)
	C5.1	0.461 (0.112)	0.377 (0.070)	0.412 (0.058)
RWGAIM	C1.4	0.459 (0.112)	0.403 (0.078)	0.374 (0.068)
	C1.8	-0.428 (0.108)	-0.436 (0.085)	-0.377 (0.065)
	C2.4	0.537 (0.136)	0.431 (0.125)	0.369 (0.074)
	C2.8	0.548 (0.137)	0.439 (0.105)	0.380 (0.078)
	C3.6	0.470 (0.093)	0.374 (0.072)	0.388 (0.050)
	C4.4	0.441 (0.093)	0.370 (0.070)	0.362 (0.046)
	C5.1	0.432 (0.096)	0.356 (0.062)	0.405 (0.056)

Table 6 gives the proportion of false positives for the 2 methods. False positives can be linked (on the same chromosome as true QTL, C1-C5) or unlinked (on chromosomes where no QTL exist, C6-C9). The rates of false positives are the number found across all 200 simulations divided by 200. RWGAIM shows reduced rates of false positives over WGAIM. Population size again has a major impact.

One aspect of WGAIM that was not examined by Verbyla et al. (2007) was bias in the

estimated QTL sizes. This was examined in the current simulation study and the results are given in Table 7. WGAIM shows higher bias (all true QTL sizes are ± 0.378) than RWGAIM. Thus RWGAIM reduces the bias particularly for the smaller population sizes, though bias still persists. The bias at population size 200 is smaller for RWGAIM while both method exhibit little bias at population size 400. Lastly, the standard deviations are generally smaller for RWGAIM.

4 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

1. Fit a base `asreml()` model

Fit a base `asreml()` (see the **ASReml-R** package) model as in (4) but without the added marker/interval genetic information term $\boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a}$. 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 experimental 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.**

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

Read in genetic data using `read.cross()` (see the **qtl** package). 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 genotypes. Thus users should be aware that the class of the returned object from `read.cross()` needs to have the structure `c("bc","cross")`. The `"bc"` is an abbreviated form for "back-cross". It is this class structure that is checked in the proceeding steps.

`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 not converge or slowly reach convergence over an extended length of time. In these instances users may need to patient.

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

This can be done using the `wgaim` function

```
cross2int(fullgeno, 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.

2 **Co-locating markers**: If `rem.mark = TRUE` then co-locating markers across the genome are removed from the marker set. The correlated markers and how they are connected is returned as part of the final object.

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 proceeding step is a complete (i.e. no missing values) genetic data set across all linkage groups.

After the linkage map manipulation, for each chromosome, the imputed marker data matrix is returned as an element of the object. Along with this, several interval calculations are returned such as distances between markers, recombination fractions and most importantly, the interval data matrix.

The final `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 `c("bc","cross")` class for backward compatibility with other functions in the **qtl** package as well as inherits the class `"interval"` for functionality within the **wgaim** package.

4. Perform QTL analysis with `wgaim()`

```
wgaim(baseModel, phenoData, intervalObj, merge.by = NULL, gen.type = "interval",
      method = "random", 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 2 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 2. The `intervalObj` contains the imputed genetic marker and interval data obtained from a call to `cross2int()` in the previous step. Thus `intervalObj` must be of class `"interval"`.

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 groups is combined and used instead.

The character string `merge.by` is then used to identify the appropriate column of `phenoData` and the newly formed whole genome genetic object with which to merge the two data sets. This merging differs depending on the 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.**

`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`.

The `method` argument in `wgaim()`

This argument is probably one of the most important arguments in the `wgaim()` call. In the current version two choices are available.

If `method = "fixed"` the `wgaim.asreml()` code uses the legacy algorithm that was available in pre-1.0 versions of **wgaim**. This version is the original algorithm discussed in Section 2 and in more detail in Verbyla et al. (2007). Users need to be aware that this algorithm uses the chromosome outlier statistic to hone its selection of a QTL at each iteration. It is now known (as discussed in Section 2.2.1) that using the chromosome

statistic is flawed when there are small linkage groups. The problem dissipates as the linkage group becomes larger. This method also places the selected QTL in the fixed effects of the linear mixed model as the algorithm proceeds (hence the `"fixed"` argument). The final set of QTL is then tested using an appropriate wald test as discussed in section 2.3.1

If `method = "random"` the `wgaim.asreml` code uses the updated algorithm and is available with versions 1.0+ of **wgaim**. In this version the selected QTL are placed in the random part of the linear mixed model as the algorithm proceeds as discussed in Verbyla & Taylor (2011) and Section 2.2. This is known to reduce bias of the selected QTL effects. This algorithm also only uses an interval outlier statistic to guide the selection of each QTL and therefore is much more appropriate if the genetic map being used contains small linkage groups.

Note: Both of these methods allow high dimensional genetic components to be added to the wgaim call through `intervalObj`.

5. Summarise QTL with various method functions

```
summary(object, intervalObj, LOD = TRUE, ...)
print(x, intervalObj, ...)
tr(object, iter = 1:length(object$QTL$effects), diag.out = TRUE, ...)
link.map(object, 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 function can be used to summarize and diagnostically check the QTL obtained from a `wgaim()` analysis. The `summary()` function assesses the significance of the QTL effects (fixed or random) and retrieves genetic marker information. For an `"interval"` analysis, chromosome, interval and distance of the left and right flanking markers are outputted. For a `"marker"` analysis, chromosome, distance of the nearest marker are displayed. `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 matrix that shows the significance of the QTL 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 some added flexibility for colouring of QTL regions as well as colour and size of printed text for all components of the map.

5 Examples

5.1 A quick example: The Zinc Data

The zinc data is available in **wgaim** as a usable date set to display the functionality of the package. The data consists of 200 observations of zinc concentration and shoot length for a DH population of wheat. There are two replicates of 90 double haploid lines from a crossing of the wheat varieties Cascades and Rac875-2 and ten each of the parents in an `id` variable. The data also includes a `Type` variable to distinguish the parents from the DH lines. The experiment also contained two blocks in a variable called `Block`.

A suitable base model for shoot length is explored by considering (4) without the random regression effects, $\boldsymbol{Z}_g\boldsymbol{M}_E\boldsymbol{a}$, attributed to genetic markers/intervals. As we are interested in the genetic variance associated with the DH lines, `Type` is modelled as a fixed effect ($\boldsymbol{\tau}$) to ensure the removal of the genetic effect associated with the parents. The `Block` is a random effect of non-interest ($\boldsymbol{u}_e$) and `id` as a set of polygenic random effects $\boldsymbol{p}$.

```
R> data("zinc", package = "wgaim")
R> sh.fm <- asreml(shoot ~ Type, random = ~Block + id, data = zinc)
```

A simple summary of the variance parameters in the model can be achieved with

```
R> summary(sh.fm)$varcomp
```

```
                   gamma  component  std.error   z.ratio constraint
Block!Block.var 0.1902258 0.03721561 0.05539823 0.6717835   Positive
id!id.var       9.1030840 1.78091965 0.28195227 6.3163871   Positive
R!variance      1.0000000 0.19563915 0.02674723 7.3143694   Positive
```

The summary reveals there is only a small difference between Blocks. However, the genetic variance is more than nine times the residual variance of the model making the response an ideal candidate for QTL analysis.

wgaim is prepackaged with a genetic map associated with the zinc data. The data has already been read in using `read.cross()` and can be loaded using

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

Alternatively to illustrate the use of `read.cross()` in conjunction with this package the same data is available from the `extdata` directory of the package library as a CSV file. A subset of the data from the CSV file is given in Table 8. 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 `AB` and missing values are `"-"`. Thus a call to `read.cross()` is

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
1	id			DH01	DH02	DH03	DH04	DH05	DH06	DH07
2	wmc469	1A1	0.00	AB	AA	AB	-	AB	AB	AB
3	wPt.5914	1A1	7.26	AB	AA	AB	AB	AA	AB	AB
4	wPt.0751	1A1	8.88	-	AA	AB	AB	AA	AB	AB
5	P42.M49.235	1A1	20.11	AA	AA	AB	AB	AA	AB	AB
6	bcd304C	1A2	0.00	AA	AB	AB	AA	AA	AA	AA
7	P31.M55.148	1A2	31.52	AA	-	AB	AA	AA	AA	AA
8	barc213	1A2	42.60	AA	AB	AB	AA	AA	AA	AA
9	P34.M48.83	1A2	43.84	AA	AB	AB	AA	AA	AA	AA
10	gwm99	1A2	54.30	AA	AA	AA	AA	AA	AA	AA

Table 8: A subset of the genetic data from the comma delimited file raccas.csv.

```
R> wgpath <- system.file("extdata", package = "wgaim")
R> raccas <- read.cross("csvr", file="raccas.csv", genotypes=c("AA","AB"),
+    dir = wgpath, na.strings = c("-", "NA"))
R> class(raccas)

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

The returned object has the required class structure.

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

```
R> names(raccas$geno)

 [1] "1A1" "1A2" "1B"  "1D1" "1D2" "2A1" "2A2" "2A3" "2B"  "2D1" "2D2" "3A1"
[13] "3A2" "3A3" "3A4" "3D1" "3D2" "4A1" "4A2" "4A3" "4B"  "4D1" "4D2" "5A1"
[25] "5A2" "5A3" "5B"  "5D1" "5D2" "6A1" "6A2" "6B1" "6B2" "6D1" "6D2" "6D3"
[37] "7A1" "7A2" "7A3" "7B"  "7D1" "7D2"
```

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(raccas$geno$"1B")

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

For each linkage group, "data" contains to the actual marker data matrix, converted into R/**qtl** format (AA = 1, AB = 2, missing values = NA). Marker names are placed as the column names and the genotype lines (DH001, DH002, ...) are the row names.

```
R> raccas$geno$"1B"$data[1:8,1:8]
```

```
     gwm18 gwm11 P40.M54.358 bcd338 P37.M48.217 wPt.7030 wPt.3457 gwm413
DH01     1     2           2      2           2        2        2      2
DH02     1     1           1      1           1        1        1      1
DH03     1     2           2      2           2        2        2      2
DH04     2     1           1      1           1       NA        1      1
DH05     1     1           1      1           1        1        1      1
DH06     1     1           1      1           1       NA        1      1
DH07     1     1           1      1           1       NA        1      1
DH08     1     1           2      2           2        2        2      2
```

The "map" element contains the map distances that were estimated using the Lander-Green hidden markov algorithm (Lander & Green, 1987) during the read.cross() process.

```
R> raccas$geno$"1B"$map
```

```
      gwm18       gwm11 P40.M54.358      bcd338 P37.M48.217    wPt.7030
    0.00000    26.48862    54.73602    57.35473    65.13091    74.86437
   wPt.3457      gwm413 P34.M58.126       barc8    wPt.2838 Clone117500
   76.34429    83.66295    92.58413    93.91230    96.80020    98.29318
     wmc406      gwm374 P37.M48.136 P34.M50.124  P40.M56.75 P39.M47.204
  100.09763   110.71761   112.82703   115.99925   121.75534   167.12669
    bcd808A P31.M60.286      gwm403 P39.M50.180    wPt.1770    wPt.6802
  174.27171   175.41956   194.79081   238.51156   244.04986   245.39910
 P44.M54.74 P39.M50.114    wPt.4290     ksuI27B     cdo393B      gwm140
  268.40321   275.31194   281.08452   297.39544   303.74771   308.84706
```

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

We can now convert the "cross" object to an "interval" object using

```
R> raccas <- cross2int(raccas, missgeno = "Mart", id = "id", rem.mark = TRUE)
R> summary(raccas)
R> class(raccas)

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

After removing coincident markers, the summary shows there is total of 458 markers across 40 linkage groups. It also reveals that just over 5% of markers were missing and imputed using the rules of Martinez & Curnow (1992). The classes of raccas 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(raccas$geno$"1A1")
```

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

Thus for each linkage group, "data" and "map" are as before. "dist" contains 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> raccas$geno$"1A1"$imputed.data[1:8,1:8]
```

```
     gwm18 gwm11 P40.M54.358 bcd338 P37.M48.217   wPt.7030 wPt.3457 gwm413
DH01     1    -1          -1     -1          -1 -1.0000000       -1     -1
DH02     1     1           1      1           1  1.0000000        1      1
DH03     1    -1          -1     -1          -1 -1.0000000       -1     -1
DH04    -1     1           1      1           1  0.9971324        1      1
DH05     1     1           1      1           1  1.0000000        1      1
DH06     1     1           1      1           1  0.9971324        1      1
DH07     1     1           1      1           1  0.9971324        1      1
DH08     1     1          -1     -1          -1 -1.0000000       -1     -1
```

and "intval" contains the interval data based on the midpoint pseudo-interval calculation of Verbyla et al. (2007)

```
R> raccas$geno$"1A1"$intval[1:6,1:6]
```

```
         gwm11 P40.M54.358     bcd338 P37.M48.217   wPt.7030   wPt.3457
DH01 0.0000000  -0.9742251 -0.9997715  -0.9979892 -0.9968539 -0.9999270
DH02 0.9772501   0.9742251  0.9997715   0.9979892  0.9968539  0.9999270
DH03 0.0000000  -0.9742251 -0.9997715  -0.9979892 -0.9968539 -0.9999270
DH04 0.0000000   0.9742251  0.9997715   0.9979892  0.9954246  0.9984933
DH05 0.9772501   0.9742251  0.9997715   0.9979892  0.9968539  0.9999270
DH06 0.9772501   0.9742251  0.9997715   0.9979892  0.9954246  0.9984933
```

The genetic object is now ready to be used in wgaim QTL analysis. For this analysis we will use the calculated genetic intervals or "intval" components of each linkage group

```
R> zn.qtlI <- wgaim(sh.fm, phenoData = zinc, intervalObj = raccas,
+     merge.by = "id", na.method.X = "include", gen.type = "interval",
+     method = "fixed")
R> summary(zn.qtlI, raccasM, LOD = FALSE)
```

```
  Chromosome Left Marker dist(cM) Right Marker dist(cM)   Size z.ratio  Pr(z)
1        3D2        gdm8    31.51       gdm136    32.64  0.436    4.13      0
2         4B     Rht1mut     54.8         gwm6    70.12   0.54    4.86      0
3        4D1     barc098        0   P42.M49.70     1.13  0.422    3.63  3e-04
4        4D2    wPt.2573        0   Rht2W.type    23.48 -0.383   -2.82 0.0048
```

The analysis reveals four significant QTL in four linkage groups. Verbyla et al. (2007) recommends the use of p-values, rather than the commonly used LOD scores, 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.

As the `gen.type` used in this analysis was `"interval"`, the summary displays the left and right marker information for the interval. We can do a similar analysis using the markers by changing the argument for `gen.type`.

```
R> zn.qtlM <- wgaim(sh.fm, phenoData = zinc, intervalObj = raccas,
+     merge.by = "id", na.method.X = "include", gen.type = "markers",
+     method = "fixed")
R> summary(zn.qtlM, raccasM)
```

```
  Chromosome      Marker dist(cM)   Size z.ratio Pr(z)   LOD
1        3D2        gdm8    31.51  0.450   4.928 0.000 5.273
2         4B     Rht1mut     54.8  0.454   5.027 0.000 5.487
3        4D1     barc098        0 -1.194  -2.728 0.006 1.616
4        4D1  P42.M49.70     1.13  1.554   3.618 0.000 2.842
5        4D2  Rht2W.type    23.48 -0.470  -4.188 0.000 3.808
6        7D1 P42.M54.113        0 -0.332  -3.571 0.000 2.770
```

Now the summary information only contains the markers that were selected during the analysis. The argument `LOD = TRUE` argument is the default to `summary.wgaim()`.

5.2 Example 2: Sunco-Tasman data

This example stresses the importance of modelling extraneous variation to a ensure a more appropriate QTL analysis. The Sunco-Tasman data is available in the `data` directory of **wgaim** and contains the results of a field trial conducted in the year 2000 with 175 double haploid lines from a crossing of wheat varieties Sunco and Tasman. The original field trial was arranged in a 31 rows by 12 columns with two replicates of each line. A milling experiment was then performed which replicated 23% of the field samples producing 456 samples milled over 38 mill days with 12 samples per day. The focus is on the trait milling yield.

```
R> stpheno <- asreml.read.table(paste(wgpath,"\\stpheno.csv", sep =""),
+    header = TRUE, sep = ",")
R> names(stpheno)
```

```
 [1] "X"        "Expt"     "Type"     "id"       "Range"    "Row"
 [7] "Rep"      "Millday"  "Milldate" "Millord"  "myield"   "lord"
[13] "lrow"
```

Smith et al. (2006) provides a phenotypic analysis of the data. They give a base model of the form

```
R> st.fmF <- asreml(myield ~ Type + lord + lrow, random = ~ id + Rep +
+    Range:Row + Millday, rcov = ~ Millday:ar1(Millord),
+    data = stpheno, na.method.X = "include")
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
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 model realistically accounts for extraneous plot variation occurring in the field as well as variation due to the design components of the milling experiment. The `lord` and `lrow` components of the fixed model are mean centred covariates of Millord and Row that capture the natural linear trends that occur in the samples across milling order on any given day and across rows in the field. 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 = stpheno,
+    na.method.X = "include")
```

The genetic map consists of 287 unique markers across 21 chromosomes and can be read in and converted using

```
R> stmap <- read.cross("csv", file="stgenomap.csv", genotypes=c("A","B"),
+    dir = wgpath, na.strings = c("-", "NA"))
R> stmap <- cross2int(stmap, missgeno="Bro", id = "id")
R> names(stmap$geno)
```

```
 [1] "1A" "1B" "1D" "2A" "2B" "2D" "3A" "3B" "3D" "4A" "4B" "4D" "5A" "5B"
[15] "5D" "6A" "6B" "6D" "7A" "7B" "7D"
```

It is possible to view the genetic map using `link.map()`. The function allows sub-setting according to distance (cM) and/or chromosome. Figure 1 shows the genetic map resulting from the first of these two commands

```
R> link.map(stmap, cex = 0.5, marker.names = "dist")
R> link.map(stmap, cex = 0.5, marker.names = "markers")
```

For larger maps a more aesthetic plot is reached by adjusting the character expansion (`cex`) parameter and increasing the plotting window width manually. You can also subset the map by distance and chromosome if you wish

```
R> link.map(stmap, cex = 0.5, marker.names = "markers",
+      chr = names(nmar(stmap)[1:12]), max.dist = 200)
```

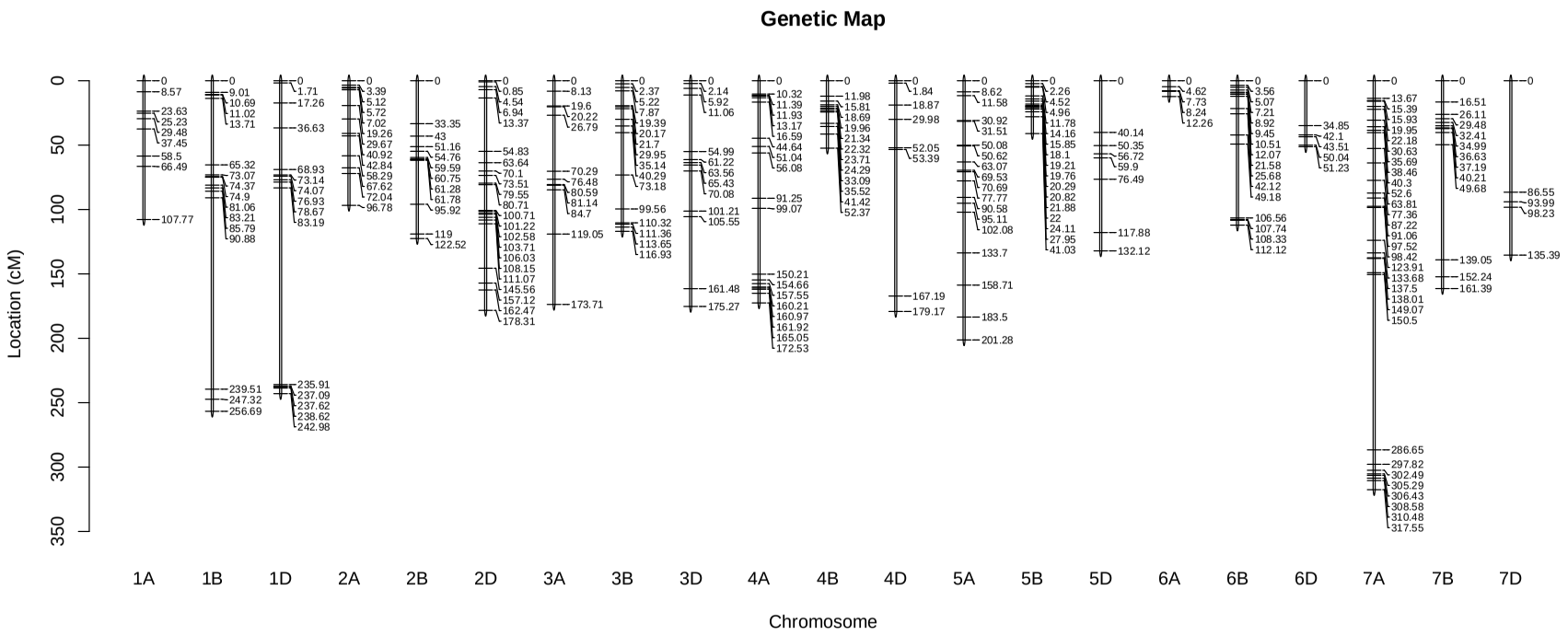


Figure 1: The genetic map for the Sunco-Tasman data. Names of chromosomes are given at the bottom and genetic distances between markers are placed alongside each of the chromosomes.

5.2.1 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`.

```
st.qtlN <- wgaim(st.fmN, phenoData = stpheno, intervalObj = stmap,
+      merge.by = "id", gen.type = "interval", method = "fixed",
+      na.method.X = "include", trace = "nullmodel.txt", verboseLev = 1)
st.qtlF <- wgaim(st.fmF, phenoData = stpheno, intervalObj = stmap,
+      merge.by = "id", gen.type = "interval", method = "fixed",
+      na.method.X = "include", trace = "fullmodel.txt", verboseLev = 1)
```

The process of selecting QTL is determined from the outlier statistics calculations in Section 2. These are saved for each QTL selection and can be viewed using the `out.stat()` command. For the first two iterations of the process the chromosome and interval outliers statistics given in Figure 2 are produced with

```
R> out.stat(st.qtlF, stmerge, int = FALSE, iter = 1:2, cex = 0.6,
+    ylim = c(0, 6.5))
R> out.stat(st.qtlF, stmerge, int = TRUE, iter = 1:2, cex = 0.6)
```

For each iteration the interval outlier statistics are plotted across the whole genome according to their distance and separated by differentiable colours according to their chromosome. There is also an additional argument that allows the user to subset the genetic map to specific chromosomes which is only available when `int = TRUE`.

```
R> out.stat(st.qtlF, stmerge, int = TRUE, iter = 1:5, cex = 0.6,
+    chr = c("2B","6B","5A","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   7D.2   4D.1   4B.1  1B.13   6B.5  5A.13  1B.1  4A.2  3D.5
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
```

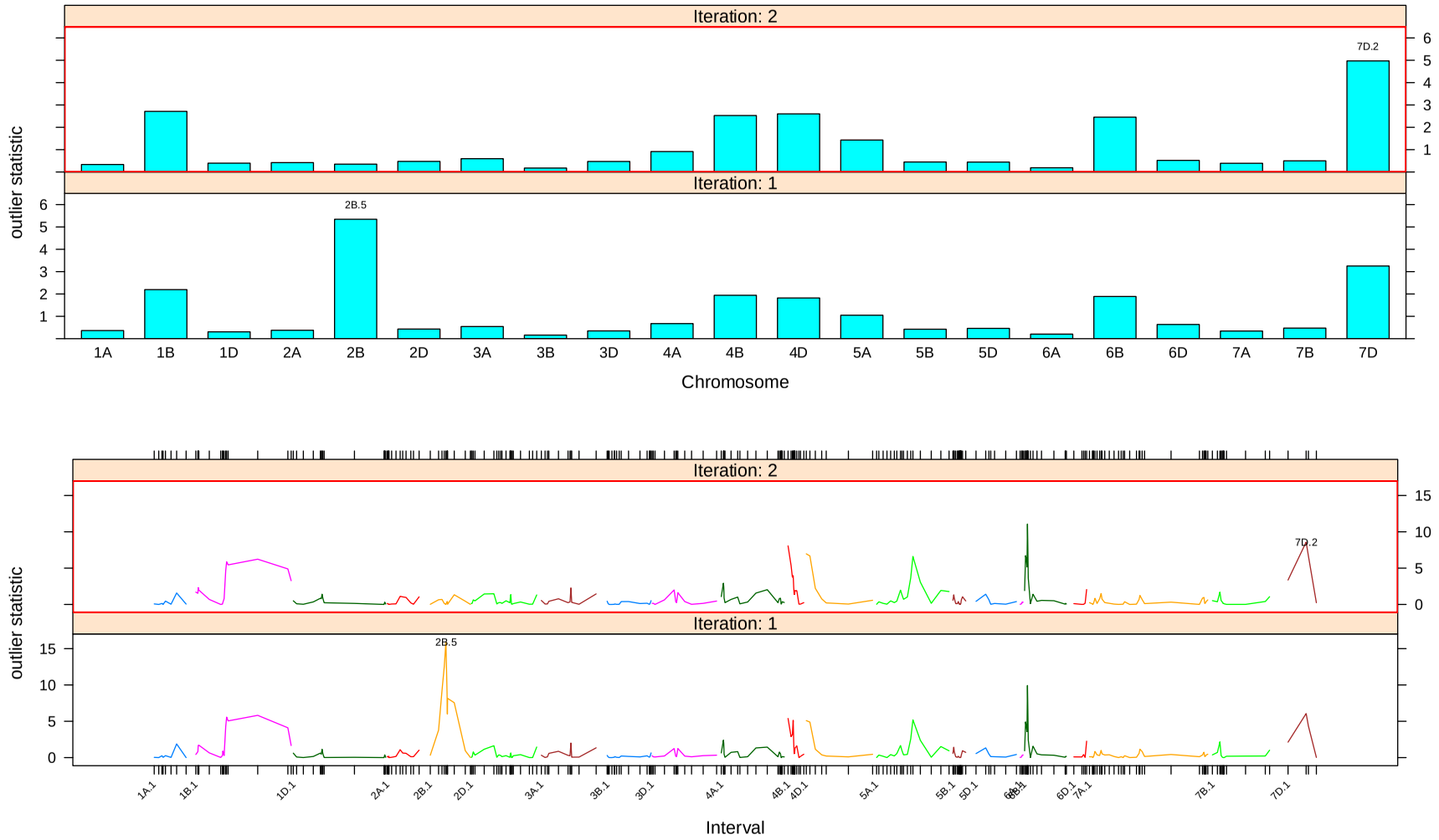


Figure 2: Chromosome and interval outlier statistics for the first two iterations of the wgaim fit for the full model.

```
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 0.021
Iter.9  <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 0.008 0.009
Iter.10 <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 <0.001 0.005 0.008 0.013

Outlier Detection Diagnostic.
============================
              L0       L1 Statistic Pvalue
Iter.1  -309.563 -250.669   117.787 <0.001
Iter.2  -279.483 -243.213    72.541 <0.001
Iter.3  -272.049 -240.762    62.574 <0.001
Iter.4  -269.746 -238.879    61.734 <0.001
Iter.5  -256.599 -235.993    41.211 <0.001
Iter.6  -251.322 -232.332     37.98 <0.001
Iter.7  -236.172 -225.886    20.572 <0.001
Iter.8  -225.186   -221.5     7.372  0.003
Iter.9  -225.029 -221.577     6.904  0.004
Iter.10  -223.37 -221.047     4.647  0.016
Iter.11 -223.008  -220.78     4.456  0.017
Iter.12 -221.138 -220.029     2.219  0.068
```

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 4B.1. The second display presents the likelihood ratio tests, $-2\log\mathbf{\Lambda}$, for the significance of the QTL variance parameter, γ_a, in (4), with the inclusion of the last hypothesis test where the null model is retained.

5.2.2 Visualising your QTL results

Full summaries are available throught the usual `summary.wgaim()` command

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

```
   Chromosome Left Marker dist(cM) Right Marker dist(cM)   Size z.ratio Pr(z)
1          1B      gwm550        0    P36.M67.1     9.01  0.182   2.905 0.004
2          1B       gwm11    90.88       gwm140   239.51 -0.804  -5.920 0.000
3          2A      wmc198    29.67       wmc170    40.92 -0.201  -3.117 0.002
4          2B   wmc474USQ    54.76       wmc35a    59.59  0.828  13.283 0.000
5          3D   TeloPAGG2    54.99    TeloPAGG1    61.22 -0.162  -2.670 0.008
6          4A      germin    10.32       cdo795    11.39  0.162   2.700 0.007
7          4B     barc193        0        csME1    11.98 -0.447  -7.104 0.000
8          4D    Rht2.mut        0        csME2     1.84  0.318   5.353 0.000
9          5A   PAACTelo2    95.11    P46.M37.4   102.08 -0.348  -5.697 0.000
10         6B      cdo507     8.92      barc354     9.45 -0.455  -7.748 0.000
11         7D      gwm437    86.55        wmc94    93.99  0.283   4.592 0.000
```

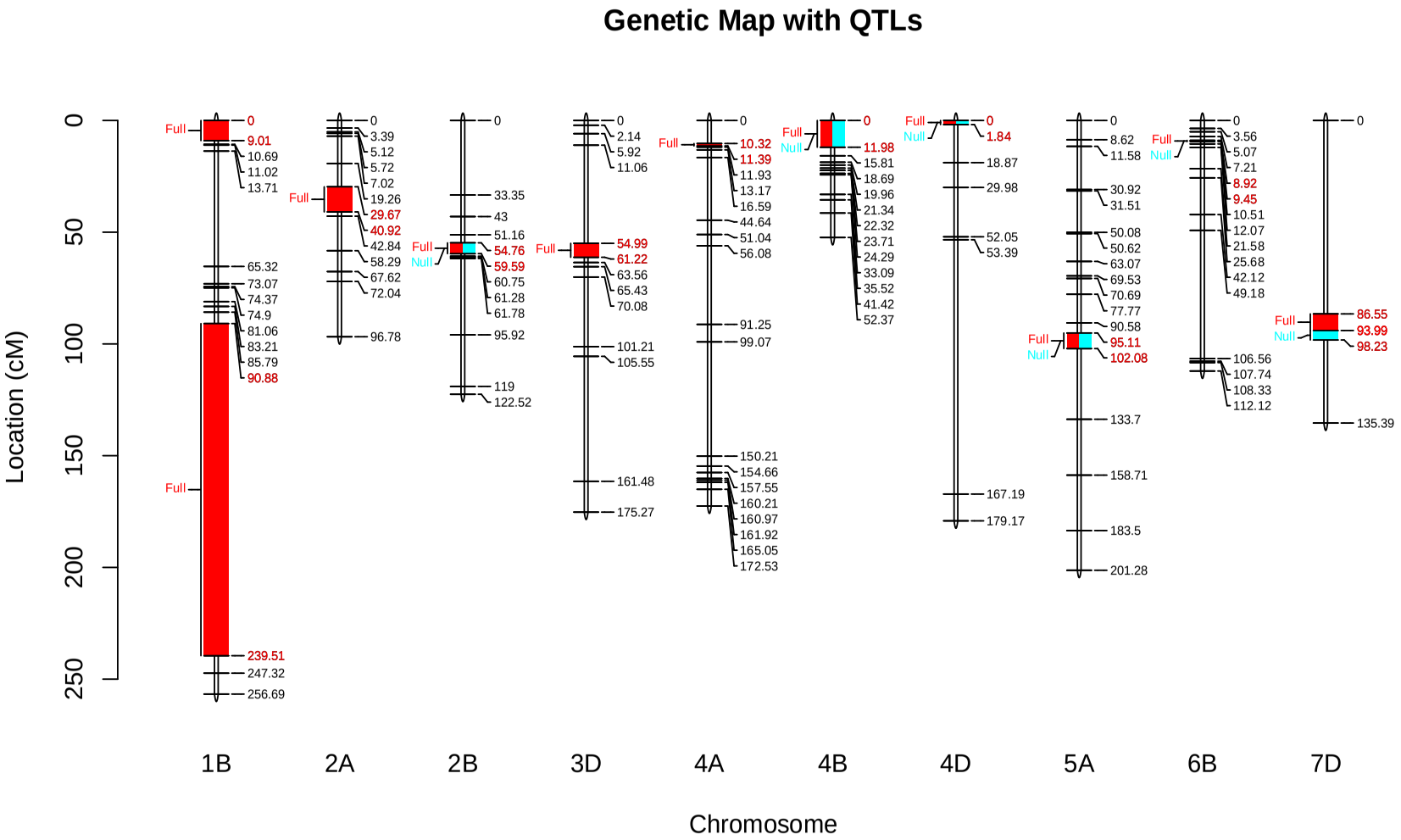


Figure 3: Genetic map with QTL for the Full and Null models obtained from an analysis of the Sunco-Tasman data. Markers and intervals for the QTL are highlighted and trait names are placed on the left hand side of the chromosomes.

The summary actually 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**.

```
R> qtlTable(st.qtlF, st.qtlN, intervalObj = stmap, labels = c("Full",
+    "Null"), columns = 1:7)
```

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, stmap, marker.names = "dist", cex = 0.6,
+    trait.labels = "Full")
```

Multiple models or traits can be handled through `link.map.default()`. For example, Figure 3 is produced with

```
R> link.map.default(list(st.qtlF, st.qtlN), stmap, 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-Tasman data and can be seen in Figure 4. These have been produced using the following criteria: Changing 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), stmap, marker.names = "dist",
+    trait.labels = c("Full", "Null"), list.col = list(q.col = c("turquoise",
+    "salmon"), m.col = "red", t.col = c("turquoise", "salmon")), col = "gray")
```

A monochromatic plot with increased sizes for the trait labels.

```
R> link.map.default(list(st.qtlF, st.qtlN), stmap, 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")
```

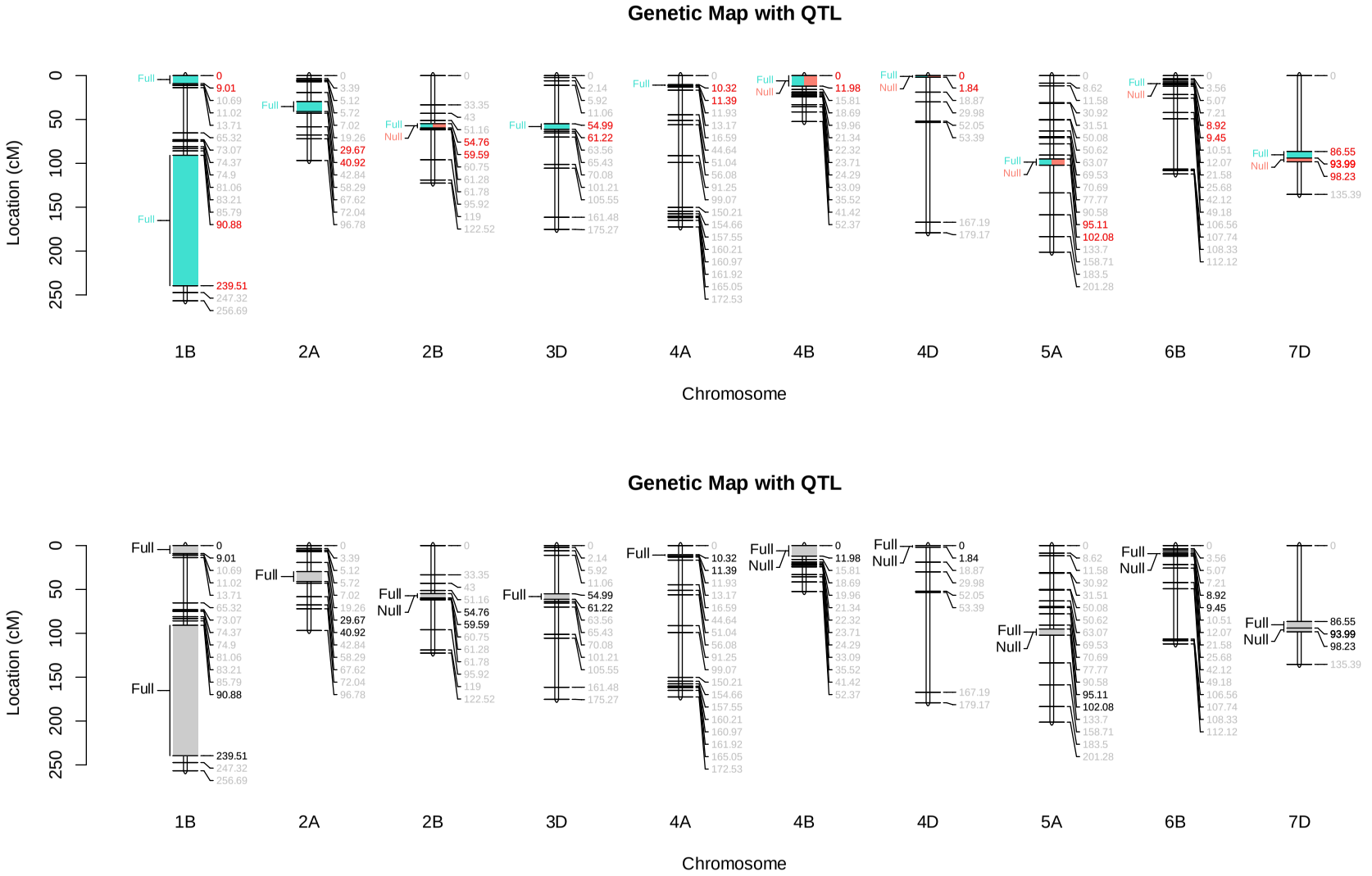


Figure 4: Customized genetic maps with QTL for the full and Null models obtained from an analysis of the Sunco-Tasman data.

5.2.3 Marker analysis

Similar to the previous example a whole genome marker QTL analysis of the Sunco-Tasman data can be performed by changing the `gen.type` argument.

```
R> st.qtlFR <- wgaim(st.fmF, phenoData = stpheno, intervalObj = stmap,
+      merge.by = "id", gen.type = "marker", method = "random",
+      na.method.X = "include", trace = "fullmodel.txt", verboseLev = 1)
```

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

```
R> summary(st.qtlFR, stmap, LOD = TRUE)
```

```
 Chromosome    Marker dist(cM)   Size  Prob    LOD
1          1B     cdo473    85.79 -0.263 1.000  4.070
2          1B    ksuI27a   247.32 -0.233 1.000  3.295
3          2A     wmc170    40.92 -0.198 0.999  2.346
4          2B wmc474USQ    54.76  0.363 0.998  1.749
5          2B     wmc35a    59.59  0.408 0.999  2.207
6          4B      csME1    11.98 -0.423 1.000 10.600
7          4D  Rht2.mut        0  0.285 1.000  4.991
8          5A PAACTelo2    95.11 -0.332 1.000  6.608
9          6B   barc354     9.45 -0.438 1.000 11.553
10         7D     wmc94    93.99  0.280 1.000  4.561
```

As discussed previously, for the random case the p-values are replaced by probabilties and for whole genome marker regression the summary contains only the closest marker that influences the trait. It is interesting to note that adjacent markers are chosen on 2B whereas the interval analysis found one large QTL in the interval between these two markers.

The random method only calculates interval outlier statistics and for this analysis places the outlier statistics at the marker position. The outlier statistics for the first first five iterations can be seen in Figure 5

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

Fitting the Null model in a similar manner.

```
R> st.qtlNR <- wgaim(st.fmN, phenoData = stpheno, intervalObj = stmap,
+      merge.by = "id", gen.type = "marker", method = "random",
+      na.method.X = "include", trace = "nullmodel.txt", verboseLev = 1)
```

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 6. The QTL are now highlighted with plotting symbols that can be altered with the usual arguments, `pch` and `cex`.

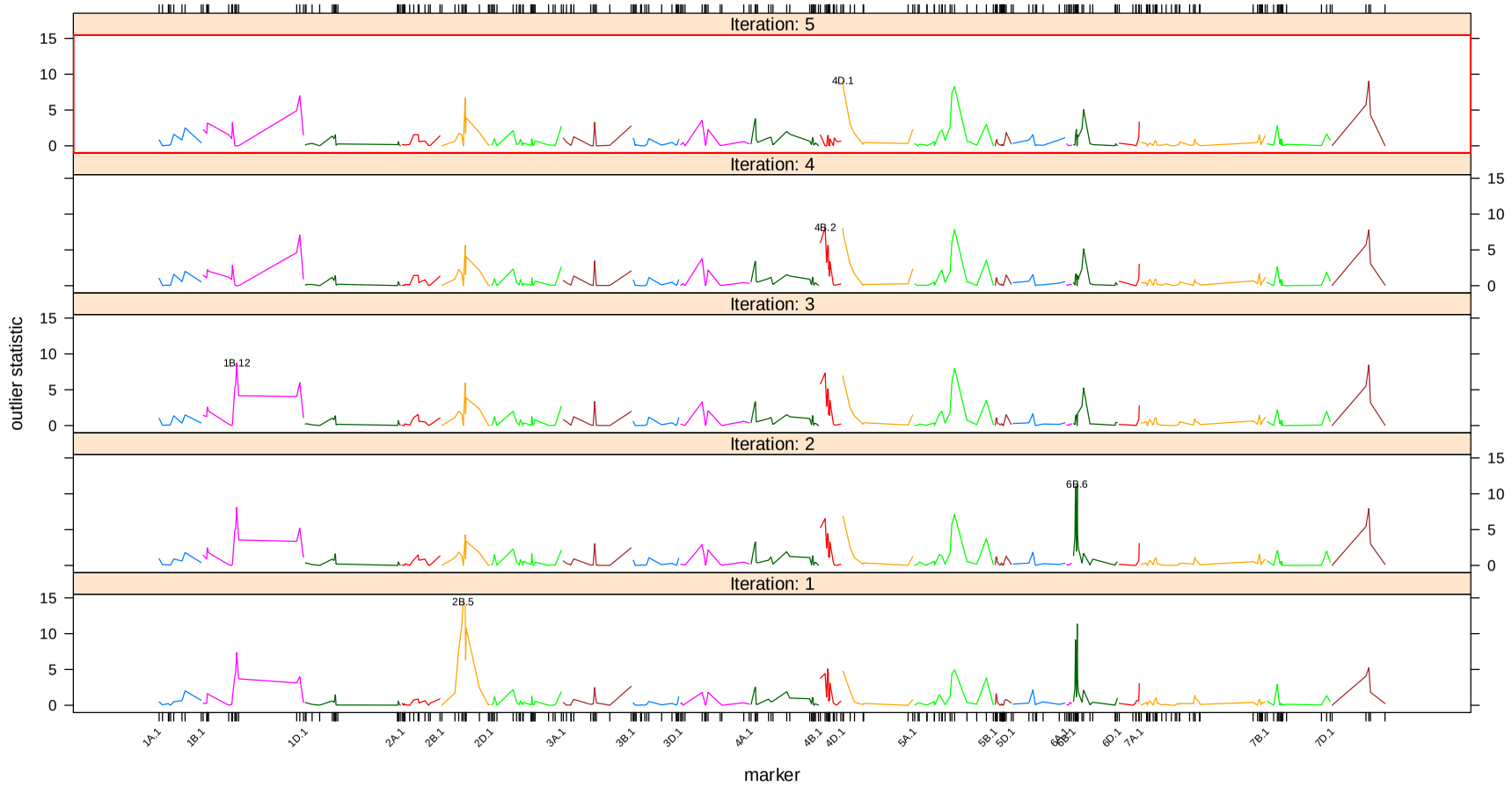


Figure 5: Outlier statistic plots for the first five iterations of the full marker regression model using the random WGAIM method

```
R> link.map.default(list(st.qtlFR, st.qtlNR), stmap, 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)
```

5.3 A quick high dimensional example

This section illustrates the accelerated performance of the WGAIM approach when the genetic component required for analsyis is high dimensional. A high dimensional genetic marker data set can be created from an existing marker map using the simulation function, `sim.geno()` in the R/**qtl** package. In this example we use the Sunco-Tasman linkage map.

Firstly, the Sunco-Tasman data is re-read in using `read.cross()`

```
R> stmap <- read.cross("csv", file="stgenomap.csv", genotypes=c("A","B"),
+     dir = wgpath, na.strings = c("-", "NA"))
sum(nmar(stmap))
```

The original data has 287 markers across 21 chromosomes. An extended map can be created with

```
R> tempmap <- sim.geno(stmap, n.draws = 1, step = 3)
```

The function uses the same hidden markov model that estimated the original distances to infill the chromosomes with new simulated markers at distances specified by the user (`step = 3`). By default `sim.geno()` returns the extended map in a separate list element of the genetic object. Some trickery is needed to convert the object into a useable form

```
R> newmap$geno <- lapply(tempmap$geno, function(el){
+     el$data <- el$draws
+     el$data <- matrix(el$data, nrow = nrow(el$data))
+     el$map <- attr(el$draws, "map")
+     el$draws <- NULL
+     el})
R> nmar(newmap)
```

```
 1A  1B  1D  2A  2B  2D  3A  3B  3D  4A  4B  4D  5A  5B  5D  6A  6B  6D  7A  7B
 44 101  95  45  52  81  69  54  71  76  30  67  85  30  52   9  53  23 136  66
 7D
 50
```

This produces a total of 1289 markers spanning th 21 chromosomes. This can now be converted to an "`interval`" object and analysed using `wgaim`.

```
R> newmap <- cross2int(newmap, rem.mark = FALSE)
R> system.time(st.qtlHD <- wgaim(st.fmF, phenoData = stpheno, intervalObj = newmap,
```

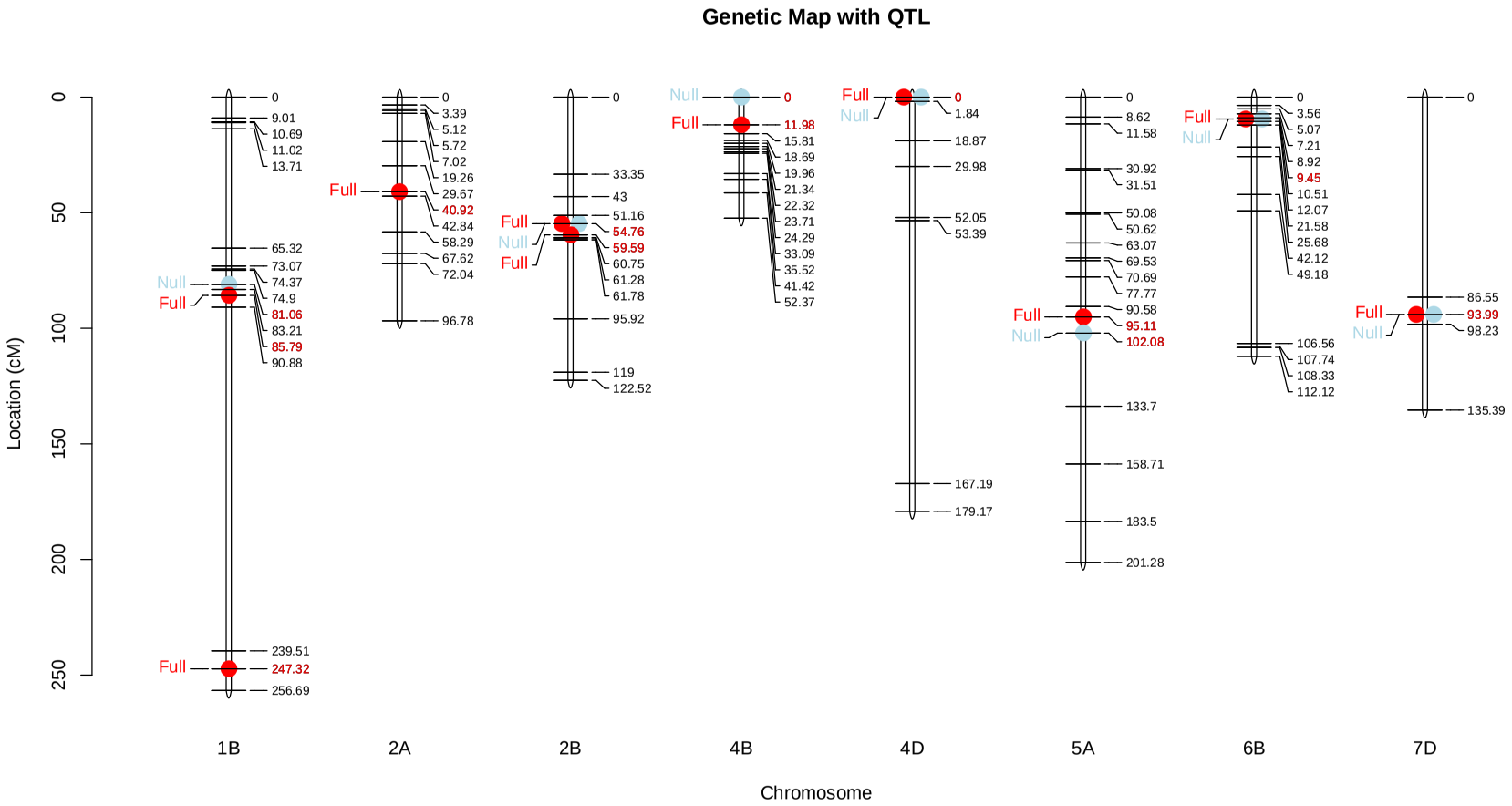


Figure 6: Genetic map with QTL for the Full and Null models. Marker QTL are highlighted according to user specifications.

```
+     merge.by = "id", gen.type = "interval", method = "fixed",
+     na.method.X = "include", trace = TRUE, verboseLev = 0))

   user  system elapsed
 137.83    1.62  140.24
```

Checking the timing for the QTL model `st.qtlF` from the previous section

```
R> system.time(st.qtlHD <- wgaim(st.fmF, phenoData = stpheno, intervalObj = stmap,
+     merge.by = "id", gen.type = "interval", method = "fixed",
+     na.method.X = "include", trace = TRUE, verboseLev = 0))

 user  system elapsed
 132.16    2.32  134.80
```

Each of the analyses finds 11 QTL and the high dimensional analysis is completed with only a negligible increase in CPU time. As outlined in Section 2.2.3, this is due to the substantial reduction in the dimension of the genetic component. In this analysis, 1289 columns of marker scores are transformed to 175 columns (i.e. the number of lines of the genetic marker set that have phenotypic information) which are then passed to the `asreml` fit during each iteration of the `wgaim` analysis.

Bibliography

Ball, R. (2010). **BayesQTLBIC**: *Bayesian Multi-Locus QTL Analysis Based on the BIC Criterion*. R package version 1.0-1, URL http://www.CRAN.R-project.org/src/contrib/Archive/bayesQTLBIC/.

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*. American Seed Trade Association, Washington, DC, 250–266.

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

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

Broman, K. W. & Speed, T. P. (2002). A model selection approach for the identification of quantitative trait loci in experimental crosses. *Journal of the Royal Statistical Society, Series B* 64 641–656.

Broman, K. W. & Wu, H. (2010). **qtl**: *Tools for Analayzing QTL Experiments*. R package version 1.15-15, URL http://www.CRAN.R-project.org/src/contrib/Archive/qtl/.

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*. Ph.D. 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. Tech. rep., University of Adelaide, Biometrics.

Golub, G. & van Loan, C. (1996). *Matrix Computations*. The Johns Hopkins University Press: London, 3rd ed.

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, URL http://dx.doi.org/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. URL http://www.genetics.org/cgi/content/abstract/178/3/1709.

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.

Meuwissen, T. H. E., Hayes, B. J. & Goddard, M. E. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics* 157 1819–1829.

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. URL http://bioinformatics.oxfordjournals.org/content/18/2/339.abstract.

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 vvaluation 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.

Taylor, J. & Verbyla, A. (2011a). R package wgaim: QTL analysis in bi-parental populations using linear mixed models. *Journal of Statistical Software* 40. URL http://www.jstatsoft.org/v40/i07.

Taylor, J. D. & Verbyla, A. P. (2011b). A variable selection method involving complex linear mixed models. *Annals of Applied Statistics* 5 000–000.

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. & Cullis, B. R. (2010). Multivariate whole genome average interval mapping: QTL analysis for multiple traits and/or multiple environments. *Theoretical and Applied Genetics* Submitted.

Verbyla, A. P., Hackett, C. A., Newton, A. M., Taylor, W. B. T. & Cullis, B. R. (2010). Multi-treatment QTL analysis using whole genome average interval mapping. *Theoretical and Applied Genetics* Submitted.

Verbyla, A. P. & Taylor, J. D. (2011). High-dimensional whole genome average interval mapping and a random effects formulation. *Theoretical and Applied Genetics, Submitted.* .

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.

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. URL http://www.jstatsoft.org/v32/b05.