National Institute for Applied Statistics Research Australia

University of Wollongong, Australia

Working Paper

09-22

On Model Based Design of Comparative Experiments in R

David Butler and Brian Cullis

National Institute for Applied Statistics Research Australia, University of Wollongong, Wollongong NSW 2522, Australia T: +61 2 42215076. E: karink@uow.edu.au

On Model Based Design of Comparative Experiments in R

David Butler
University of Wollongong
NSW

Brian Cullis
University of Wollongong
NSW

Abstract

The design of experiments with qualitative treatment effects where the observations (and possibly the treatment effects) are correlated is of interest in the context of agricultural field experiments. In the evaluation of genetic lines in Australian plant improvement programs, it is widely accepted that methods of analysis accounting for spatial trend and genetic relatedness have been found to produce accurate predictions of both additive and total genetic effects. Motivated by this framework, we propose a general method to generate optimal categorical designs under the linear mixed model, and describe a model based paradigm for design specification. The methodology is generic, with application to the design of categorical experiments in other settings, and is implemented in an R package: odw. The model based approach of odw extends the utility of R model formulae, where variance models for random factors and constructed model terms are specified using a functional style.

Keywords: optimal design, comparative experiments, correlated treatment effects, linear mixed model.

1. Introduction

Members of the broad class of block designs with categorical treatments are widely used in many statistical applications. For example, those commonly used in plant breeding programs include complete or incomplete block, row-column and α-designs (Williams and John 1996), while more recently partially replicated designs (Cullis, Smith, and Coombes 2006a) are widely accepted in early generation testing in Australian plant breeding programs. Design methods for spatially correlated data have been of interest in this setting (Butler, Eccleston, and Cullis 2008; Cullis *et al.* 2006a; Williams, John, and Whitaker 2006; Chan 1999; Martin and Eccleston 1997; Martin 1986). Few theoretical results exist and numerical optimization

has been used to construct A-optimal designs (Coombes 2002; Chan 1999; Eccleston and Whitaker 1999; Martin and Eccleston 1997), assuming fixed treatment effects and a pre-specified residual correlation model.

Design methods where there is a known correlation structure among treatment effects (for example, genetic lines) have received comparatively little attention. Cullis *et al.* (2006a), for example, consider random, though uncorrelated genetic effects in constructing partially replicated designs. The numerator relationship matrix (NRM, $\boldsymbol{A}$) gives an *average* measure of relatedness among individuals, based on identity by descent probabilities (see Lynch and Walsh 1998, for example). Alternative identical in state measures of relatedness based on molecular information are also available (Wang 2002; Pagnacco and Jansen 2001) for use in predicting genetic potential. In a small study using incomplete block designs and simple genetic models for $\boldsymbol{A}$, Bueno Filho and Gilmour (2003) showed that using the information on genetic relatedness can improve the design of such experiments. Cullis, Smith, Cocks, and Butler (2020) report the results of a simulation study that illustrates the gains in accuracy for the prediction of total genetic effects (and therefore selection) that can be achieved from model-based designs using genetic relatedness, bespoke non-genetic and residual models. The only practical restriction in the methodology presented here is that $\boldsymbol{A}^{-1}$ must exist.

In general, algorithms that optimise a design criterion such as the A- or D-value Kiefer (1974), proceed by iteratively permuting the allocation of treatments to experimental units under the supervision of an optimisation strategy. A-optimality minimises the average pairwise variance of all elementary treatment contrasts, and D-optimality minimises the average variance of parameter estimates. A-optimality is generally considered an appropriate design criterion when all treatment comparisons are of equal interest, such as genetic selection experiments. However, computing the $\mathcal{A}$-criterion for a design involves an expensive matrix inversion, and repetitive brute force calculation in a design search is not feasible. In practice, a scheme to update a matrix inverse such as described by Martin and Eccleston (1992) must be used (Coombes 2002; Chan 1999) to achieve realistic compute times.

The odw package provides a flexible model based interface that explicitly links the underlying analytical linear model to the design phase of an experiment. Computationally, we document an efficient updating scheme based on Martin and Eccleston (1992) to compute the $\mathcal{A}$-criterion for general cases where treatment effects are correlated. The discussion is structured as follows: Section 2 introduces the general linear model underlying the methodology; Section 3 outlines a computational framework for optimal design based on the mixed model equations; Section 4 introduces a model based approach to design specification through the **odw** function and includes several examples that demonstrate **odw** in practice; concluding remarks and further developments are discussed in Section 6.

2. The linear mixed model and design

With $\boldsymbol{y}$ as the $n \times 1$ vector of observations, we consider the linear mixed model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e} \tag{1}$$

with a convenient partitioning scheme where $\boldsymbol{\tau} = [\boldsymbol{\tau}_1^\top \ \boldsymbol{\tau}_2^\top]^\top$, $\boldsymbol{\tau}_1 = [\boldsymbol{\tau}_{11}^\top \ \boldsymbol{\tau}_{12}^\top]^\top$ are vectors of fixed effects of size $n_\tau \times 1$ and n_{τ_1} respectively with $n_\tau = n_{\tau_1} + n_{\tau_2}$ and $n_{\tau_1} = n_{\tau_{11}} + n_{\tau_{12}}$, $\boldsymbol{u} = [\boldsymbol{u}_1^\top \ \boldsymbol{u}_2^\top]^\top$, $\boldsymbol{u}_1 = [\boldsymbol{u}_{11}^\top \ \boldsymbol{u}_{12}^\top]^\top$ are vectors of random effects of size $n_u \times 1$ and n_{u_1} respectively

with $n_u = n_{u_1} + n_{u_2}$ and $n_{u_1} = n_{u_{11}} + n_{u_{12}}$, $\boldsymbol{e}$ is the $n \times 1$ vector of errors, and $\boldsymbol{X} = [\boldsymbol{X}_1\ \boldsymbol{X}_2] = [\boldsymbol{X}_{11}\ \boldsymbol{X}_{12}\ \boldsymbol{X}_2]$ and $\boldsymbol{Z} = [\boldsymbol{Z}_1\ \boldsymbol{Z}_2] = [\boldsymbol{Z}_{11}\ \boldsymbol{Z}_{12}\ \boldsymbol{Z}_2]$ are associated design matrices. The partitioning distinguishes between *permute* and *static* effects being those associated with the design search and those associated with terms addressing the plot structure of the experiment (including covariates) respectively. The partition of permute effects is required to define objective and linked effects. Objective effects contribute to the design optimality criterion, while linked effects are permuted in the design search but do not contribute to the design optimality criteria. To be strictly correct, the design search involves an interchange of the rows of $\boldsymbol{W}$, and the permute set of effects are those effects which are associated with the columns of $\boldsymbol{W}$ which are interchanged, while the static set of effects are those which are associated with the columns of $\boldsymbol{W}$ which are not interchanged. For the remainder of the paper we refer to these sets as the permute set and the static set. Subsets of the permute set are referred to as the objective and linked sets. By convention the overall mean parameter is in the objective set, unless otherwise stated.

Equation (1) can be written succinctly as

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{\beta} + \boldsymbol{e} \tag{2}$$

where $\boldsymbol{W} = [\boldsymbol{W}_1\ \boldsymbol{W}_2]$, $\boldsymbol{W}_1 = [\boldsymbol{X}_1\ \boldsymbol{Z}_1]$, $\boldsymbol{W}_2 = [\boldsymbol{X}_2\ \boldsymbol{Z}_2]$, $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^\top\ \boldsymbol{\beta}_2^\top]^\top$, $\boldsymbol{\beta}_1 = [\boldsymbol{\tau}_1^\top\ \boldsymbol{u}_1^\top]^\top$ and $\boldsymbol{\beta}_2 = [\boldsymbol{\tau}_2^\top\ \boldsymbol{u}_2^\top]^\top$.

The random effects $\boldsymbol{u}$ and errors $\boldsymbol{e}$ in (1) are assumed normally distributed such that

$$\begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{e} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \right)$$

where $\boldsymbol{G}_1$, $\boldsymbol{G}_2$ and $\boldsymbol{R}$ are positive definite matrices assumed to be functions of vectors of variance parameters $\boldsymbol{\sigma}_{g_1}$, $\boldsymbol{\sigma}_{g_2}$ and $\boldsymbol{\sigma}_r$ respectively. Model-based design requires values for these parameters so in the following they are regarded as known.

2.1. Mixed model equations for the set of permute effects

The mixed model equations (MME) (Robinson 1991) for (2) can be written succinctly as

$$\boldsymbol{C}\tilde{\boldsymbol{\beta}} = \boldsymbol{W}^\top \boldsymbol{R}^{-1} \boldsymbol{y} \tag{3}$$

and are given by

$$\begin{bmatrix} \boldsymbol{W}_1^\top \boldsymbol{R}^{-1} \boldsymbol{W}_1 + \boldsymbol{G}_1^* & \boldsymbol{W}_1^\top \boldsymbol{R}^{-1} \boldsymbol{W}_2 \\ \boldsymbol{W}_2^\top \boldsymbol{R}^{-1} \boldsymbol{W}_1 & \boldsymbol{W}_2^\top \boldsymbol{R}^{-1} \boldsymbol{W}_2 + \boldsymbol{G}_2^* \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\beta}}_1 \\ \tilde{\boldsymbol{\beta}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_1^\top \boldsymbol{R}^{-1} \boldsymbol{y} \\ \boldsymbol{W}_2^\top \boldsymbol{R}^{-1} \boldsymbol{y} \end{bmatrix} \tag{4}$$

where $\boldsymbol{G}^* = \begin{bmatrix} \boldsymbol{G}_1^* & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_2^* \end{bmatrix}$, $\boldsymbol{G}_1^* = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_1^{-1} \end{bmatrix}$ and $\boldsymbol{G}_2^* = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_2^{-1} \end{bmatrix}$.

It follows that the reduced MME for the permute effects $\tilde{\boldsymbol{\beta}}_1$ are given by

$$\boldsymbol{C}_{11}\tilde{\boldsymbol{\beta}}_1 = \boldsymbol{W}_1^\top \boldsymbol{P}_2 \boldsymbol{y} \tag{5}$$

where $\boldsymbol{C}_{11} = \boldsymbol{W}_1^\top \boldsymbol{P}_2 \boldsymbol{W}_1 + \boldsymbol{G}_1^*$, $\boldsymbol{P}_2 = \boldsymbol{R}^{-1} - \boldsymbol{R}^{-1}\boldsymbol{W}_2(\boldsymbol{W}_2^\top \boldsymbol{R}^{-1}\boldsymbol{W}_2 + \boldsymbol{G}_2^*)^- \boldsymbol{W}_2^\top \boldsymbol{R}^{-1}$ and $(\boldsymbol{W}_2^\top \boldsymbol{R}^{-1}\boldsymbol{W}_2 + \boldsymbol{G}_2^*)^-$ is any particular generalised inverse of $\boldsymbol{W}_2^\top \boldsymbol{R}^{-1}\boldsymbol{W}_2 + \boldsymbol{G}_2^*$. It can be shown that

$$\boldsymbol{P}_2 = \boldsymbol{V}_2^{-1} - \boldsymbol{V}_2^{-1}\boldsymbol{X}_2(\boldsymbol{X}_2^\top \boldsymbol{V}_2^{-1}\boldsymbol{X}_2)^- \boldsymbol{X}_2^\top \boldsymbol{V}_2^{-1}$$

where $\boldsymbol{V}_2 = \boldsymbol{Z}_2\boldsymbol{G}_2\boldsymbol{Z}_2^\top + \boldsymbol{R}$ and $(\boldsymbol{X}_2^\top\boldsymbol{V}_2^{-1}\boldsymbol{X}_2)^-$ is any particular generalised inverse of $\boldsymbol{X}_2^\top\boldsymbol{V}_2^{-1}\boldsymbol{X}_2$. The matrix $\boldsymbol{P}_2$ has rank $n - \text{rank}(\boldsymbol{X}_2)$ and is unique, and it is the Moore-Penrose inverse of $\boldsymbol{T} = \boldsymbol{M}_2\boldsymbol{V}_2\boldsymbol{M}_2$, where $\boldsymbol{M}_2 = \boldsymbol{I}_n - \boldsymbol{X}_2(\boldsymbol{X}_2^\top\boldsymbol{X}_2)^-\boldsymbol{X}_2^\top$. That is $\boldsymbol{T} = \boldsymbol{P}_2^+$.

2.2. Prediction and optimal design criteria

The aim is to find an optimal or near optimal design with respect to a n_π-vector of estimable functions $\boldsymbol{\pi} = \boldsymbol{D}\boldsymbol{\beta}_1$ where $\boldsymbol{D}$ is a known matrix with $n_{w_1} = n_{\tau_1} + n_{u_1}$ columns. The vector of estimable functions, $\boldsymbol{\pi}$, involves only objective effects, but may involve fixed, random or both fixed and random effects. Gilmour, Cullis, Welham, Gogel, and Thompson (2004) provide a computationally efficient algorithm for forming predictions from the linear mixed model specified in (2). For brevity in the following we use the terminology of Gilmour *et al.* (2004) and refer to $\boldsymbol{\pi}$ as the vector of predictions, which we assume are estimable. Gilmour *et al.* (2004) provide a simple test of estimability of predictions which fit naturally into their prediction algorithm. Briefly, given $\boldsymbol{D}$, the vector of predictions and associated prediction error variance/covariance matrix $\boldsymbol{\Lambda}$ are formed by recursive absorption from an extended set of mixed model equations (see Robinson 1991, for example). Full details can be found in Gilmour *et al.* (2004).

For known $\boldsymbol{\sigma}_{g_1}, \boldsymbol{\sigma}_{g_2}$ and $\boldsymbol{\sigma}_{\boldsymbol{r}}$

$$\boldsymbol{D}(\boldsymbol{\beta}_1 - \tilde{\boldsymbol{\beta}}_1) \sim \text{N}(\boldsymbol{0},\ \boldsymbol{\Lambda}) \tag{6}$$

where $\boldsymbol{\Lambda} = \boldsymbol{D}\boldsymbol{C}_{11}^-\boldsymbol{D}^\top$ and $\boldsymbol{C}_{11}^-$ a particular generalised inverse of the coefficient matrix of (5).

A commonly used optimality criterion is $\mathcal{A}$-criterion, which is equivalent to minimising the average pairwise variance of all elementary contrasts $(\pi_i - \pi_j), i \neq j$. The $\mathcal{A}$-criterion ($\mathcal{A}$) is usually considered appropriate in circumstances where all treatments are of equal interest, such as early stage plant breeding trials (Martin 1986). Bueno Filho and Gilmour (2007) developed a Bayesian design criterion for selection experiments in plant breeding based on a utility function that minimizes the risk of an incorrect selection. They show that this is in fact the $\mathcal{A}$-criterion on the prediction error variance (PEV) matrix for the vector of random entry (treatment) effects. Bueno Filho and Gilmour (2003) and Cullis *et al.* (2006a) use this criterion for generating optimal or near-optimal designs for plant breeding designs when the treatments are correlated, and for partially replicated designs. In this case it can be shown that

$$\mathcal{A} = \sum_i \sum_{j<i} \text{pev}(\tilde{\pi}_i - \tilde{\pi}_j)/n_\pi/(n_\pi - 1)$$

where pev refers to the prediction error variance of its scalar (or matrix) argument. A computational form for $\mathcal{A}$ is given by

$$\mathcal{A} = \frac{2}{n_\pi - 1}(\text{tr}(\boldsymbol{\Lambda}) - \frac{1}{n_\pi}\boldsymbol{1}_{n_\pi}^\top\boldsymbol{\Lambda}\boldsymbol{1}_{n_\pi}).$$

3. Finding an optimal design

Finding an optimal design in categorical experiments is a problem in combinatorial optimization, where n entities must be allocated to n locations such as with the Quadratic Assignment Problem or the Travelling Salesman Problem. Here, optimal design seeks a permutation of

the given set of treatments of interest to the experimental units that is optimal with respect to a predefined statistical criterion. Formally, if $\boldsymbol{\Omega}$ is the n-vector of experimental units, a *design* is a function $\mathcal{F}$ such that unit ω_j is allocated treatment $\mathcal{F}(\omega_j)$ (see Bailey 2008b). For a given design matrix $\boldsymbol{W}$, say, $\mathcal{F}(\omega)$ allocates treatment effect $\boldsymbol{\beta}[i]$ to ω_j, where i is the position of the non-zero element of $\boldsymbol{W}_j^\top$, the j^{th} row vector of $\boldsymbol{W}$. In practice, we consider a design as a permutation vector $\boldsymbol{p}$ ordering the rows of $\boldsymbol{W}$, with a notional $n \times n$ permutation matrix $\boldsymbol{\mathcal{P}}$. Reordering the rows of $\boldsymbol{\mathcal{P}}$ provides a convenient mechanism to dynamically alter the mapping returned by $\mathcal{F}(\omega)$.

Let $\mathcal{D}$ be the set of designs attainable through all possible (and permissible) permutations of the rows of $\boldsymbol{\mathcal{P}}$. Even for modest designs it is generally impossible to enumerate all solutions in $\mathcal{D}$, and an efficient evaluation strategy directed by an optimisation algorithm is necessary. Given a suitable starting design and choice of optimality criterion, $\mathcal{A}$, optimal design search methods share a common set of features in exploring the design space $\mathcal{D}$, that include:

1. A method to calculate $\mathcal{A}$ for a given $\boldsymbol{\mathcal{P}}$,
2. An interchange policy to move to a neighbouring configuration in $\mathcal{D}$,
3. A sampling strategy for $\mathcal{D}$ and acceptance policy for new configurations,
4. A stopping rule to terminate the search.

An interchange strategy is a perturbation function $S()$ operating on the rows of $\boldsymbol{\mathcal{P}}$. A straightforward interchange function for $S()$ is typically chosen which simply swaps two rows of $\boldsymbol{\mathcal{P}}$, subject to any resolvability or other practical constraints. Computationally, a design is represented in a permutation vector $\boldsymbol{p}$ and a new permutation $\boldsymbol{p}^*$ is generated as $\boldsymbol{p}^* = s(\boldsymbol{p})$, where $s()$ is the corresponding vector perturbation function.

3.1. Search algorithm

Items 2-4 above encapsulate the search algorithm in the optimization process. The search algorithm implemented in **odw** is based on the the tabu search first reported in Glover (1989, 1990). Simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983) has been used in optimal design problems (Chan 1999; Elliot, Eccleston, and Martin 1999) but Coombes (2002) found its performance inferior to tabu based methods.

The tabu search implements the the robust method of Taillard (1991) and the suggestion of Woodruff and Zemel (1993) to use hashing vectors to maintain the *tabu* list. A strict cycle avoidance scheme that encodes $\boldsymbol{p}$ into a 64 bit hash code using the FNV hash (Fowler, Knoll, Vo, and Eastlake 2011) is used. This tabu strategy is relatively simple to implement, and the results of Randall and Abramson (1997) suggest that storing the full solution in the tabu list coupled with a small probability of accepting an inferior solution was very competitive when compared across a range of assignment problems.

3.2. Computation

In the most general case, the prediction eror variance matrix $\boldsymbol{\Lambda}$ can be computed directly from augmenting the mixed model equations (4) with $\boldsymbol{D}$ (Gilmour *et al.* 2004). Succinctly,

using the notation of (3) the extended mixed model equations are

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{D} \\ \boldsymbol{D}^\top & \boldsymbol{C}_{11} \end{bmatrix}, \tag{7}$$

and absorbing $\boldsymbol{C}_{11}$ gives $\boldsymbol{Q}^* = -\boldsymbol{D}\boldsymbol{C}_{11}^-\boldsymbol{D}^\top$, that is, $\boldsymbol{\Lambda} = -\boldsymbol{Q}^*$. This approach is amenable to the formation and inclusion of a general $\boldsymbol{D}$ in the search for an optimal design. A prediction design matrix pre-processor is not yet available and special cases of $\boldsymbol{D}$ are implemented through various arguments to the **odw** function described in Section 4; the numerical methods as used in **odw** are described here.

The rows of $\boldsymbol{W}_2$ are considered invariant and the permutation $\boldsymbol{p}$ operates only on the rows of $\boldsymbol{W}_1$, noting that not all columns of $\boldsymbol{W}_1$ need contribute to the calculation of $\mathcal{A}$. At iteration $\iota = 0$, $\boldsymbol{\mathcal{P}}^{[0]} = \boldsymbol{I}_n$ and $\boldsymbol{p}^{[0]} = [1, 2, \ldots, n]$. Computing $\mathcal{A}$ by brute force involves forming $\boldsymbol{\Lambda}^{[\iota+1]} (= (\boldsymbol{C}_{11}^-)^{[\iota+1]})$ for each update $\boldsymbol{W}_1^{[\iota+1]} = \boldsymbol{\mathcal{P}}^{[\iota+1]}\boldsymbol{W}_1^{[\iota]}$, where $\boldsymbol{\mathcal{P}}^{[\iota+1]} = S(\boldsymbol{\mathcal{P}}^{[\iota]})$. This is computationally prohibitive for all but small designs.

Martin and Eccleston (1992) developed an updating method for finding optimal designs under a linear model with correlated errors. Their algorithm is extended here, but allowing the more general setting of correlated objective effects, within the framework of a linear mixed model. The interchange of two rows of $\boldsymbol{W}_1$ is equivalent to first removing two rows from Ω, followed by adding the two units back, but in reverse order. Martin and Eccleston (1997) suggest using a four-step approach which involves adding or removing one unit to obtain the new $\boldsymbol{C}_{11}^-$. Chan (1999) presented a two-step approach which she claims to be simpler and easier to implement. We begin by developing a two-step approach, similar to that proposed by Chan (1999), but a four-step approach is also presented as this has proven to be competitive to the two-step approach in terms of computational load. The four-step approach is also more suitable for use in the context of early stage trials, where the presence of *singleton* treatments (that is, those treatents which occur on only one plot) can cause computational problems as discussed by Coombes (2002).

Let the current design contain $n = r + s$ units, which are referred to as plots in the following. Consider the retention of r plots by removing s plots, and both $\boldsymbol{W}_1$ and $\boldsymbol{P}_2$ are partitioned conformably with the removal of s plots as follows

$$\boldsymbol{W}_1 = \begin{bmatrix} \boldsymbol{W}_{1r} \\ \boldsymbol{W}_{1s} \end{bmatrix} \quad \text{and} \quad \boldsymbol{P}_2 = \begin{bmatrix} \boldsymbol{P}_{2;rr} & \boldsymbol{P}_{2;rs} \\ \boldsymbol{P}_{2;sr} & \boldsymbol{P}_{2;ss} \end{bmatrix}$$

where for example $\boldsymbol{W}_{1r}$ is the design matrix for the subset of the design corresponding to the r retained plots so has r rows and c_{w_o} columns. The matrix $\boldsymbol{T}$ is also partitioned conformably with $\boldsymbol{P}_2$. We note that in many cases $\boldsymbol{X}_2$ is null in which case $\boldsymbol{P}_2 = \boldsymbol{V}_2^{-1}$ and hence $\boldsymbol{T} = \boldsymbol{V}_2$. It follows that the coefficient matrix of the reduced MMEs for the subset of the design corresponding to the r retained plots is

$$\boldsymbol{C}_{11}^{(r)} = \boldsymbol{W}_{1r}^\top \boldsymbol{T}_{rr}^\sim \boldsymbol{W}_{1r} + \boldsymbol{G}_1^*$$

where $\boldsymbol{T}_{rr}^\sim$ is any particular generalised inverse of $\boldsymbol{T}_{rr}$.

Using a similar argument to Martin and Eccleston (1992) and results on the inverse of partitioned matrices (see for example Searle 1982) it can be shown that

$$\boldsymbol{C}_{11} - \boldsymbol{C}_{11}^{(r)} = \boldsymbol{F}_{rs}\boldsymbol{P}_{2;ss}^-\boldsymbol{F}_{rs}^\top \tag{8}$$

where $\boldsymbol{F}_{rs} = \boldsymbol{W}_{1r}^{\top}\boldsymbol{P}_{2;rs} + \boldsymbol{W}_{1s}^{\top}\boldsymbol{P}_{2;ss}$. Hence an updating formula for $\boldsymbol{C}_{11}^{(r)-}$ can now be derived using (8) as follows. Using Corollary 18.2.15 in Harville (1997) (p.432)

$$\boldsymbol{C}_{11}^{(r)-} = \boldsymbol{C}_{11}^{-} + \boldsymbol{C}_{11}^{-}\boldsymbol{H}_{rs}(\boldsymbol{P}_{2;ss} - \boldsymbol{H}_{rs}^{\top}\boldsymbol{C}_{11}^{-}\boldsymbol{H}_{rs})^{-}\boldsymbol{H}_{rs}^{\top}\boldsymbol{C}_{11}^{-} \tag{9}$$

where $\boldsymbol{H}_{rs} = \boldsymbol{F}_{rs}\boldsymbol{P}_{2;ss}^{-}\boldsymbol{P}_{2;ss}$. If $\boldsymbol{P}_{2;ss}$ is non-singular then $\boldsymbol{H}_{rs} = \boldsymbol{F}_{rs}$. Furthermore, if $(\boldsymbol{P}_{2;ss} - \boldsymbol{H}_{rs}^{\top}\boldsymbol{C}_{11}^{-}\boldsymbol{H}_{rs})$ is non-singular then (9) becomes

$$\boldsymbol{C}_{11}^{(r)-} = \boldsymbol{C}_{11}^{-} + \boldsymbol{C}_{11}^{-}\boldsymbol{H}_{rs}(\boldsymbol{P}_{2;ss} - \boldsymbol{H}_{rs}^{\top}\boldsymbol{C}_{11}^{-}\boldsymbol{H}_{rs})^{-1}\boldsymbol{H}_{rs}^{\top}\boldsymbol{C}_{11}^{-} \tag{10}$$

Next we consider adding s units back to the design, but with the appropriate permutation applied to the rows of $\boldsymbol{W}_{1s}$. In the simple case of $s = 2$, then provided that the interchange is legal, then the two rows of $\boldsymbol{W}_{1s}$ are interchanged. The new design matrix for the full set of $n = r + s$ units is therefore given by

$$\boldsymbol{W}_{1}^{*} = \begin{bmatrix} \boldsymbol{W}_{1r} \\ \boldsymbol{W}_{1s}^{*} \end{bmatrix}$$

where $\boldsymbol{W}_{1s}^{*}$ is the permuted design matrix associated with the s units.

Using a similar approach to the deletion of r units, we consider the coefficient matrix of the reduced MMEs for the full, new design with $r + s$ units which is given by

$$\boldsymbol{C}_{11}^{(r+s)} = \boldsymbol{W}_{1}^{*\top}\boldsymbol{P}_{2}\boldsymbol{W}_{1}^{*} + \boldsymbol{G}_{1}^{*}$$

It can be shown that

$$\boldsymbol{C}_{11}^{(r+s)} - \boldsymbol{C}_{11}^{(r)} = \boldsymbol{F}_{rs}^{*}\boldsymbol{P}_{2;ss}^{-}\boldsymbol{F}_{rs}^{*\top} \tag{11}$$

where $\boldsymbol{F}_{rs}^{*} = \boldsymbol{W}_{1r}^{\top}\boldsymbol{P}_{2;rs} + \boldsymbol{W}_{1s}^{*\top}\boldsymbol{P}_{2;ss}$. Hence an updating formula for $\boldsymbol{C}_{11}^{(\boldsymbol{r+s})-}$ is

$$\boldsymbol{C}_{11}^{(r+s)-} = \boldsymbol{C}_{11}^{(r)-} - \boldsymbol{C}_{11}^{(r)-}\boldsymbol{H}_{rs}^{*}(\boldsymbol{P}_{2;ss} + \boldsymbol{H}_{rs}^{*\top}\boldsymbol{C}_{11}^{(r)-}\boldsymbol{H}_{rs}^{*})^{-}\boldsymbol{H}_{rs}^{*\top}\boldsymbol{C}_{11}^{(r)-} \tag{12}$$

where $\boldsymbol{H}_{rs}^{*} = \boldsymbol{F}_{rs}^{*}\boldsymbol{P}_{2;ss}^{-}\boldsymbol{P}_{2;ss}$. If $\boldsymbol{P}_{2;ss}$ is non-singular then $\boldsymbol{H}_{rs}^{*} = \boldsymbol{F}_{rs}^{*}$. Furthermore, if $(\boldsymbol{P}_{2;ss} + \boldsymbol{H}_{rs}^{*\top}\boldsymbol{C}_{11}^{(\boldsymbol{r})-}\boldsymbol{H}_{rs}^{*})$ is non-singular then (12) becomes

$$\boldsymbol{C}_{11}^{(r+s)-} = \boldsymbol{C}_{11}^{(r)-} - \boldsymbol{C}_{11}^{(r)-}\boldsymbol{H}_{rs}^{*}(\boldsymbol{P}_{2;ss} + \boldsymbol{H}_{rs}^{*\top}\boldsymbol{C}_{11}^{(\boldsymbol{r})-}\boldsymbol{H}_{rs}^{*})^{-1}\boldsymbol{H}_{rs}^{*\top}\boldsymbol{C}_{11}^{(r)-} \tag{13}$$

Using the two updating formulae, one for the removal and one for the addition allows us to compute the new $\mathcal{A}$-value from $\boldsymbol{\Lambda}^{(\boldsymbol{r+s})} = \boldsymbol{D}\boldsymbol{C}_{11}^{(\boldsymbol{r+s})-}\boldsymbol{D}^{\top}$.

Setting $s = 1$ leads to the four-step updating scheme proposed by Martin and Eccleston (1992). The four-step approach can have computational advantages and additionally it can be implemented to handle designs in which all of the objective effects are fixed effects and the design contains singletons. Using the updating formulae in (10) and (13), as an example, when $s = 1$ we have:

$$\boldsymbol{C}_{11}^{(r)-} = \boldsymbol{C}_{11}^{-} + \boldsymbol{C}_{11}^{-}\boldsymbol{h}_{rs}(p_{2;ss} - \boldsymbol{h}_{rs}^{\top}\boldsymbol{C}_{11}^{-}\boldsymbol{h}_{rs})^{-1}\boldsymbol{h}_{rs}^{\top}\boldsymbol{C}_{11}^{-} \tag{14}$$

$$\boldsymbol{C}_{11}^{(r+s)-} = \boldsymbol{C}_{11}^{(r)-} - \boldsymbol{C}_{11}^{(r)-}\boldsymbol{h}_{rs}^{*}(p_{2;ss} + \boldsymbol{h}_{rs}^{*\top}\boldsymbol{C}_{11}^{(r)-}\boldsymbol{h}_{rs}^{*})^{-1}\boldsymbol{h}_{rs}^{*\top}\boldsymbol{C}_{11}^{(r)-} \tag{15}$$

for removal and addition of a unit respectively where

$$\boldsymbol{W}_1 = \begin{bmatrix} \boldsymbol{W}_{1r} \\ \boldsymbol{w}_{1s}^\top \end{bmatrix}, \quad \boldsymbol{W}_1^* = \begin{bmatrix} \boldsymbol{W}_{1r} \\ \boldsymbol{w}_{1s}^{*\top} \end{bmatrix} \quad \text{and} \quad \boldsymbol{P}_2 = \begin{bmatrix} \boldsymbol{P}_{2;rr} & \boldsymbol{p}_{2;rs} \\ \boldsymbol{p}_{2;sr}^\top & p_{2;ss} \end{bmatrix}$$

and

$$\begin{aligned} \boldsymbol{h}_{rs} &= \boldsymbol{f}_{rs} = \boldsymbol{W}_{1r}^\top \boldsymbol{p}_{2;rs} + \boldsymbol{w}_{1s} p_{2;ss} \\ \boldsymbol{h}_{rs}^* &= \boldsymbol{f}_{rs}^* = \boldsymbol{W}_{1r}^\top \boldsymbol{p}_{2;rs} + \boldsymbol{w}_{1s}^* p_{2;ss} \end{aligned}$$

for $p_{2;ss} - \boldsymbol{h}_{rs}^\top \boldsymbol{C}_{11}^- \boldsymbol{h}_{rs} > 0$.

The two-step approach is straightforward, however it is useful to illustrate the four-step approach in more detail. As a simple example of the four-step approach consider interchange of two units in the design, denoted by (ω_a, ω_b). The rows of the objective design matrix $\boldsymbol{W}_1$ are indexed using Ω such that $\boldsymbol{W}_1 = \boldsymbol{W}_1[\Omega,]$ and let Ω_{-a} be the set of plots excluding ω_a and Ω_{-b} be the set of plots excluding ω_b, then the four-step approach consists of the following four steps:

1. Remove ω_a: $\boldsymbol{W}_{1r} = \boldsymbol{W}_1[\Omega_{-a},]$ and $\boldsymbol{w}_{1s}^\top = \boldsymbol{W}_1[\omega_a,]$ then use (14);
2. Add ω_b: $\boldsymbol{W}_{1r} = \boldsymbol{W}_1[\Omega_{-a},]$ and $\boldsymbol{w}_{1s}^{*\top} = \boldsymbol{W}_1[\omega_b,]$ then use (15);
3. Remove ω_b: $\boldsymbol{W}_{1r} = \boldsymbol{W}_1^*[\Omega_{-b},]$ and $\boldsymbol{w}_{1s}^\top = \boldsymbol{W}_1^*[\omega_b,]$ then use (14);
4. Add ω_a: $\boldsymbol{W}_{1r} = \boldsymbol{W}_1^*[\Omega_{-b},]$ and $\boldsymbol{W}_{1s}^{*\top} = \boldsymbol{W}_1[\omega_a,]$ then use (15)

noting that if a singleton is included in the set (ω_a, ω_b) then the steps must be arranged so that the singleton is not removed first.

4. Model based design in odw

4.1. A brief overview of odw

Design generators are often discipline specific, and seldom explicitly link the underlying linear model to the design, or subsequent analysis. For example, in agricultural field trials design factors specifying trial layout are typically referred to as *column* and *row*, while for the same class of design in a laboratory setting these may correspond to *day* and *sequence*, say. Linear mixed models can be succinctly represented by extending the symbolic model formulae of Wilkinson and Rogers (1973a). The full linear mixed model in **odw** is described with three formula objects:

fixed to specify the fixed effects terms,

random to specify the random effects, and

residual to specify the residual (error) variance structure.

The `fixed` and `random` formulae contain model terms separated by a '+' operator, with compound terms, such as interactions, formed with the ':' operator in the R convention. The `residual` formula defines the error structure with one or more model terms separated by the ':' operator; by necessity the product of the numbers levels of these factors equals the number of observational units. Variance models for random terms and constructed model factors are assigned by special model functions acting on the term(s) of interest. In this context, the ':' operator implies that the variance model for the compound term is formed as the Kronecker product of the contributing terms. The presence of the intercept in the model is implicit, but can be explicitly included or omitted using the '+1' or '-1' constructs, respectively. The `fixed` formula must be present and contain at least the intercept.

The terms appearing in the model formulae are resolved in a pre-existing **R** data frame, which serves as the *base* or starting design. This data frame is mandatory, must have the appropriate factor declarations. Initially, 1000 interchanges of the experimental units using a greedy steepest descent strategy are undertaken prior to a supervised search.

In addition to the model formulae introduced above, there are several other arguments to **odw** that complete the design specification and direct the design search. Principal among these include identifying $\boldsymbol{W}_1$, setting pre-specified values for $\boldsymbol{\sigma}_{g_1}$, $\boldsymbol{\sigma}_{g_2}$ and $\boldsymbol{\sigma}_r$, and identifying resolvability or other constraints for the interchange policy. Table 1 outlines some key arguments to **odw** with reference to the corresponding components of the linear model or methodology (where applicable).

Table 1: Some **odw** arguments additional to the model formulae. Only `permute` and `data` are mandatory.

Argument	Value	Context
`permute`	$\sim$ *formula* specifying the permute terms. This is a one sided formula with the objective term optionally separated by the "|" operator from any linked terms. These linked terms do not contribute to $\mathcal{A}$.	$\boldsymbol{W}_1$, $\boldsymbol{D}$
`swap`	$\sim$ *formula* controlling legal unit exchanges.	$S()$
`optimize`	*character vector* identifying the levels of the objective factor to include in the $\mathcal{A}$ set; defaults to those present in the data.	$\mathcal{O}$, $\boldsymbol{D}$
`group`	*list* where each component is a numeric vector specifying g contiguous fields in `data` that are to be considered as a single term. The component names can then appear in **odw** model formulae using the `grp()` special function.	$\boldsymbol{X}$, $\boldsymbol{Z}$
`start.values`	*logical* which if `TRUE` returns a list containing the default variance parameter values; these can be replaced with user specified estimates.	$\boldsymbol{G}$, $\boldsymbol{R}$
`G.param`	*list* assigning values of pre-specified variance parameters.	$\boldsymbol{G}$
`R.param`	*list* assigning values of pre-specified variance parameters.	$\boldsymbol{R}$
`criterion`	*string* specifying the optimality criterion: $\mathcal{A}$ or pev= $\mathrm{tr}(\boldsymbol{\Lambda})$.	$\mathcal{O}$
`data`	*data frame* containing the initial configuration.	

Variance models for random terms are specified with special model functions in the `random` and `residual` formulae. Table 2 summarises the variance functions available in **odw** and

additional special functions that extend the linear model.

Table 2: Summary of special model functions

Function	Description
Constructor type functions	
`at(obj, lev)`	Defines one or more binary variables with value 1 where the levels of `obj` in the data matches those in the vector `lev`, or 0 otherwise.
`dsum(form, lev)`	Used in the `residual` formula to define $\boldsymbol{R}$. The `formula` object `form` allows a conditioning factor on the right of a '|' operator that partitions $\boldsymbol{e}$ into independent sections; the expression to the left models the error structure within each section. If `lev` is absent the same variance structure is applied to all sections.
`grp(obj)`	Defines a new factor `obj` with g levels from contiguous columns within the data. The g columns of data are identified by a character or numeric vector component `obj` of the `odw()` `group` argument.
`xpr(form)`	Creates a new model factor with corresponding columns in the design matrix formed from an algebraic expression of existing model terms. The expression is given in the **R** `formula` object `form` and the levels of all participating terms must conform in size.
Variance models	
`id(obj)`	Identity model family for factor `obj`.
`ar1(obj)`	First order autoregressive correlation family.
`cor(obj)`	Simple correlation family.
`vm(obj, source)`	Includes a known variance structure for additive genetic effects in the model. The relationship matrix $\boldsymbol{G_K}$ or its inverse $\boldsymbol{G_K^{-1}}$ is given by the `source` argument.
`ide(obj, source)`	Includes a variance structure for non-additive genetic effects in the model. The factor levels are set by the known structure in `source`, and the variance model is set to `idv()` or `id()` depending on context.
`ric(obj, source)`	Includes the Ricardo variance structure $\sigma_a^2\boldsymbol{G_K} + \sigma_e^2\boldsymbol{I}$ for total genetic effects in the model.
`str(form, var)`	Applies the direct product variance structure in the formula argument `var` to the terms given in the formula `form`.

4.2. Notes on special functions

The model functions in Table 2 can be broadly classified as constructor functions or variance functions. The former are concerned with generating new model terms and may have no intrinsic variance parameters, while the latter specify a variance model to apply to a factor either singularly or as part of a direct product structure. The distinction is not unambiguous.

A correlation model $\boldsymbol{\Sigma} = [\rho_{ij}]$ where $\rho_{ii} = 1\ \forall i$ can be converted to a variance matrix by $\boldsymbol{V}^{1/2}\boldsymbol{\Sigma}\boldsymbol{V}^{1/2}$ where $\boldsymbol{V}$ is a diagonal matrix of (possibly equal) variance parameters. The

variance form of the correlation matrices is specified by appending 'v' or 'h' to the correlation function name. In the former case this yields a homogeneous variance model while the latter gives the corresponding heterogeneous model. In the following, `obj` refers to a factor in the data with n levels except where noted.

Conditional factors

A conditional factor is one that is present only when another factor has a particular level. The `at()` and `dsum()` functions specify conditional structures, albeit in different contexts.

`at(obj, lev)`

The `at()` function constructs one or more factors depending on whether the `lev` argument is a scalar or a vector. For a scalar `lev`$= \ell$, the resulting factor has 1 for data records where `obj`$== \ell$. If `lev` is a vector of length n_ℓ, then n_ℓ factors are included in the model. These factors are typically used in compound terms to allow for individual effects in a nested hierarchy, such as *blocks* within *experiments*, say. If `lev` is absent, all levels of `obj` are used; this is equivalent to `idh(obj)`.

`dsum(form, lev)`

The `dsum()` function strictly applies to the `residual` formula and allows for multiple independent *sections* within $\boldsymbol{e}$. The conditioning factor (`cf`) is specified to the right of a '|' operator in the formula `form`, and the variance model for each level of `cf` specified in `lev` is given on the left. Multiple `dsum()` functions separated by '+' may be used in a `residual` formula. Formally, `dsum()` specifies $\boldsymbol{R}$ as the direct sum of variance matrices, each of which may be the direct product of one or more dimensional factors.

Augmenting the design matrix W

`grp(obj)`

The `grp()` function augments the design matrix $\boldsymbol{W}$ with a contiguous set of columns from the data, considering them to be defined by a single factor. The companion `group` list argument to `odw()` identifies the sets of data columns that define each `grp()` factor; each string `obj` must be in the set of `names(group)`.

`xpr(form)`

The `xpr()` function forms a new set of columns in $\boldsymbol{W}$ from an arithmetic operation on existing ones, where participating columns are identified by their generating factors in an algebraic expression (**R** formula) `form`. The factors in `form` must have the same number of levels. Allowed operators are '+', '-', '*' and '/ ' with any constants or coefficients given explicitly; all other symbols are expected to resolve to model terms. For example, (xpr($\sim$ 0.5*Male + 0.5*Female)) forms a set of columns in $\boldsymbol{W}$ that are the (matrix) sum of those for `Male` and `Female` scaled by 0.5.

Identity variance models

`id(obj)`, `idv(obj)`, `idh(obj)`,

where $\boldsymbol{\Sigma} = \boldsymbol{I}$. The `id()` correlation model or the `idv()` simple variance model, depending on context, is the default for terms in the `random` or `residual` formulae where a model is not

explicitly specified.

Correlation and time series type models

`cor(obj)`, `corv(obj)`, `corh(obj)`,

where $\mathbf{\Sigma} = [\rho_{ij}], \rho_{ii} = 1\ \forall i$ and $\rho_{ij} = \rho$ for $i \neq j$. These models have 1, 2 and $1+n$ parameters, respectively.

`ar1(obj)`, `ar1v(obj)`, `ar1h(obj)`,

where $\mathbf{\Sigma} = [\rho_{ij}], \rho_{ii} = 1\ \forall i$ and $\rho_{ij} = \rho^{|i-j|}$ for $i \neq j$. These models have 1, 2 and $1 + n$ parameters, respectively.

Known relationship structures

`vm(obj, source)`, `ide(obj, source)`, `ric(obj, source)`,

where $\mathbf{\Sigma} = \boldsymbol{G_K} = [v_{ij}]$ and $v_{ij} = v_{ji}\ \forall i, j$ for `vm()` and `ric()`, and $\mathbf{\Sigma} = \boldsymbol{I}$ for `ide()`. Either $\boldsymbol{G_K}$ or $\boldsymbol{G_K^{-1}}$ or $\dim(\boldsymbol{I})$ is given by the `source` argument:

- a three column matrix holding $\boldsymbol{G_K^{-1}}$ in co-ordinate form in row major order. This triplet matrix must have a `rowNames` attribute giving the levels of the model term being defined, or
- a `matrix` (or `Matrix` object) with a `dimnames` attribute. This may be $\boldsymbol{G_K}$ or $\boldsymbol{G_K^{-1}}$; if $\boldsymbol{G_K^{-1}}$, `source` must have an attribute `INVERSE` set to `TRUE`,

noting that more levels may be defined in `source` than exist in `obj`.

The functions `vm()` and `ric()` generate the variance models $\sigma_a^2\boldsymbol{G_K}$ and $\sigma_a^2\boldsymbol{G_K} + \sigma_e^2\boldsymbol{I}$ for additive and total genetic effects, respectively. While total genetic effects can be specified in the `random` formula by `vm()+ide()`, only `ric()` is permitted in `permute` for efficiency.

The `source` argument to `ide()` is optional if `ide()` is preceded by `vm()` in the model.

General variance structures

`str(form, var)`

The `str()` function applies a variance model given in the `var` formula to the model given in the `form` formula argument. The `var` argument is a one sided formula specifying a direct product structure with variance functions separated by ':' operators. The size of the variance structures can be given as an integer argument to the variance functions in place of the usual factor object.

Typically, a variance structure applies to an individual term. Sometimes it is appropriate, for example in random regression models, to allow for covariance between terms. The columns in $\boldsymbol{W}$ corresponding to the terms in `form` are kept together, and the variance structure `var` begins at the first term and covers the subsequent terms in the sequence.

4.3. The odw class

The **odw** class includes `plot` and `summary` methods, and a call to **odw** returns an S3 object of class `'odw'` with the following components:

`design` A data frame with the `permute` terms in design order.

`permutation` The permutation vector $\boldsymbol{p}$.

`criterion` The optimality criterion $\mathcal{O}$ at termination.

`G.param` List object containing the pre-specified variance components for terms in the `random` formula.

`R.param` List object containing the pre-specified variance components for terms in the `residual` formula.

`call` The function call.

5. Examples

5.1. Generating designs

The following examples assume that the **odw** package has been loaded with

```
> library(odw)
```

prior to loading the data. The choice of pre-specified variance parameters in the examples is for illustration, and is not meant to imply their suitability in practice; in general these parameter settings would be based on prior experience or specific knowledge of the specific experimental conditions at hand. The A-optimality criterion is used in all cases.

5.2. Latinized row-column designs

This example considers the construction of latinized and t-latinized designs and are taken from John and Williams (1995) and John and Williams (1998). Table 3 presents a summary of the attributes of three designs. Designs `t1` and `t2` are presented in tables 1 and 2 of John and Williams (1998), while design `t3` is presented in table 6.6 of John and Williams (1995). Design `t1` is a 2-latinized design for 24 varieties with two replicates, design `t2` is a 2-latinized row-column design for 56 varieties with three replicates and design `t3` is a (1-)latinized row-column design for 40 varieties (provenances) with six replicates. Design `t1` was generated using a simulated annealing algorithm in which the optimality criteria was a weighted linear function of two design functions, corresponding to the resolvable block design and the long block binary design respectively. Design `t2` was constructed in two stages. Stage one generated a 2-latinized block design in blocks of size 7. In the second stage, the columns of the design (i.e. the long columns of width 2) were fixed and their algorithm carried out within column interchanges to find an efficient row-column design. Lastly, design `t3` was also generated in two stages. A latinized α-design was generated using ALPHA+, then an efficient row-column design was generated with permissable swaps being within long columns.

We begin by setting up the data-frames which contain the plot and treatment factors necessary for construction of each design. The three data frames are presented in the following. In each data frame, the original allocation of `Variety` was been altered by applying a random permutation to the elements of `Variety`.

Design	v	k	s	r	RC	tL	Ajw	Aodw
t1	24	4	6	2	n	2	1.0845850	1.0845850
t2	56	7	8	3	y	2	0.7497202	0.7494786
t3	40	5	8	6	y	1	0.3748953	0.3748950

Table 3: Summary of the attributes and optimality criteria for the t-latinized designs. Design attributes use standard notation, RC indicates the design is a row-column design and tL indicates the latinized level. Optimality criteria are presented for the designs in the literature (Ajw) and obtained from **odw** (Aodw).

```
> sapply(t1s,nlevels)

    Row     Col     Rep Longcol Variety
      8       6       2       3      24

> sapply(t2s,nlevels)

    Row     Col     Rep Longcol Variety
     21       8       3       4      56

> sapply(t3s,nlevels)

    Row     Col     Rep Variety
     30       8       6      40
```

Note that the data frame for design `t3` does not contain an additional factor for long columns, as this design is (1-)latinized.

The efficiency of each design was computed and verified using the intra-block linear model in **odw**. The terms in the model differed slightly for each design depending on the attributes of the design and the **odw** calls are presented below:

```
$t1
odw(fixed = ~Variety + Rep + Rep:Col + Longcol, permute = ~Variety,
    search = "tabu+rw", maxit = 0, data = t1)

$t2
odw(fixed = ~Variety + Rep + Rep:Col + Row + Longcol, permute = ~Variety,
    search = "tabu+rw", maxit = 0, data = t2)

$t3
odw(fixed = ~Variety + Rep + Col + Rep:Col + Row, permute = ~Variety,
    swap = ~Rep, search = "tabu+rw", maxit = 0, data = t3)
```

An additional term, `Row` was included for the row-column designs, `t2` and `t3`, while the `Longcol` term for design `t3` was simply `Col`. All terms were fitted as fixed effects and the objective and permute term was `Variety`. The $\mathcal{A}$ values for each design can be converted to efficiencies as follows:

```
> c((2*1/2),(2*1/3),(2*1/6))/
+   c('t1'=sapply(t1des.ll,function(x) x$criterion)[[1]],
+   't2'=sapply(t2des.ll,function(x) x$criterion)[[1]],
+   't3'=sapply(t3des.ll,function(x) x$criterion)[[1]])

       t1        t2        t3
0.6448598 0.7142562 0.7070759
```

noting that the default setting in **odw** for σ^2 is 1.

To construct an optimal model-based design in **odw** it is natural use a model which is as close to the model used for the analysis. The recommended method of analysis for block designs for the recovery of inter-block information, is to fit a linear mixed model using residual maximum likelihood Patterson and Thompson (1971). This linear mixed model will be referred to as the inter-block LMM. To specify this model in **odw** requires terms associated with the plot structure of the experiment be declared as `random`, while terms associated with the treatment structure, in this case `Variety`, be declared as `fixed`. In our experience, the search algorithm in **odw** is more reliable for constructing classical design by using the inter-block LMM compared with the intra-block LMM, which is the linear mixed model with all terms declared as `fixed`. Construction using the inter-block LMM is also very robust to the specification of the variance parameters for the random terms. Unless specified, the values used for the variance parameters in the design search are the default of 0.1.

Optimality criteria for the published designs under the inter-block LMM are presented in table 3. These were obtained from **odw** using a call which declared all terms associated with the plot structure as `random`, and setting `maxit` to 0. The call to produce an optimal 2-latinized design for `t1` is given by

```
desodw <- odw(fixed = ~ Variety, random=~ Rep  + Rep:Col + Longcol,
permute = ~Variety, search = "tabu+rw", data = t1s,maxit=100)
summary(desodw)$is.binary
table(with(desodw$design, table(Variety,Rep:Col)))
```

where the data-frame `t1s` has identical variables to `t1` except the elements of `Variety` have been permuted. Note that we set `maxit` to 100, but the optimal design was found in one iteration. The code chunk also illustrates the `summary` method for checking the binary properties of the design. On the other hand, optimal t-latinized row-column designs for `t2` and `t3` were obtained in two stages. Stage one used **odw** to construct an optimal t-latinized design using the following calls:

```
t2desodw <- odw(fixed = ~ Variety, random = ~ Rep + Rep:Col + Longcol,
    permute = ~Variety, search = "tabu+rw", data = t2s,maxit=100)
summary(t2desodw)$is.binary
with(t2desodw$design,table(table(Variety,Rep:Col)))
t3desodw <- odw(fixed = ~ Variety, random = ~ Rep  + Col + Rep:Col,
    permute = ~Variety, search = "tabu+rw", data = t3s,maxit=1000)
summary(t3desodw)$is.binary
with(t3desodw$design,table(table(Variety,Rep:Col)))
t3desodw <- update(t3desodw,swap=~Rep)
```

Finding an optimal t-latinized design for `t3` proved more difficult, and so an additional `update` was used in which the permissable interchanges were restricted to within replicates. This reduces the size of the design space and enhances the performance of the supervised learning algorithm for this design. Stage two involved using the design from stage one as the initial configuration, adding `Row` to the random model formula and restricting permisssable interchanges to within the intersection of `Rep` and `Longcol` and `Col` respectively. The calls were

```
t2desodw <- odw(fixed = ~ Variety,
    random = ~ Rep + Rep:Col + Row + Longcol,
    permute = ~Variety, swap = ~Rep:Longcol, search = "tabu+rw",
    data = t2desodw$design,maxit=1000)
with(t2desodw$design,table(table(Variety,Rep:Longcol)))
summary(t2desodw)$is.binary
t2desodw <- update(t2desodw,maxit=1000)
t3desodw <- odw(fixed = ~ Variety,
    random = ~ Rep  + Col + Rep:Col + Row,
    permute = ~Variety, swap = ~Rep:Col, search = "tabu+rw",
    data = t3desodw$design,maxit=500)
summary(t3desodw)$is.binary
with(t3desodw$design,table(table(Variety,Rep:Col)))
t3desodw <- update(t3desodw,maxit=5000)
```

The $\mathcal{A}$ values for the three designs constructed using **odw** are presented in table 3, matching or improving on those in the literature.

5.3. Partially replicated designs using genetic relatedness

Designs for total genetic effects

Cullis *et al.* (2006a) introduced the class of partially replicated (p-rep) designs to address challenges with the design of early stage selection experiments in plant breeding programmes. These designs incorporate a flexible strategy for replication status of genotypes in which test lines with limited seed can be sown in one plot, while other test lines and commercial varieties which have adequate seed supply can be sown in two, three or more plots. This property results in the use of the full set of genotypes rather than a subset of those with sufficient seed for two or more replicates. The p-rep design is also a sensible alternative to grid plot designs where resources are wasted by interspersing plots allocated to unreplicated test lines with a systematic grid of plots containing a few check genotypes (see Kempton (1982) and Cullis, Lill, Fisher, Read, and Gleeson (1989)). The p-rep designs advocated by Cullis *et al.* (2006a) are widely used in most Australian plant breeding programs and are gaining more popularity overseas. Efficient model-based designs have been constructed using the R software package **DiGGeR** (Coombes 2009). Recently Cullis *et al.* (2020) demonstrated that substantial gains in response to selection can be achieved by using genetic relatedness for the construction of efficient model-based p-rep designs.

In this example, we illustrate the construction of an efficient model-based design for a stage one (S1) trial grown in 2022 at Narrabri, in northern NSW, using **odw**. The experiment

involved a total of 1139 genotypes and was one of four experiments which were located across the agro-ecological area of interest. The experiment at Narrabri was the so-called "home-site", which means that every genotype in the full multi-environment trial is grown at this site. Other satellite sites typically grow a subset of the genotypes; the exact allocation of genotypes to satellite sites depends on seed supply of each genotype and land availability for each satellite site. There were a total of 1280 plots at the home-site and these were laid out in two row-adjacent rectangular lattices each comprising 20 columns and 32 rows. Thus the overall dimension of the experiment was 64 rows by 20 columns. The two rectangular arrays are referred to as co-located trials, or simply trials, and are used to facilitate sowing and harvesting operations. Hence trials are considered as a major blocking factor in the design construction process. It is clear that a p-rep design must be used so that all genotypes can be grown. Comprehensive ancestral information was available and this was used to form the NRM, $\boldsymbol{A}$. The mean genetic relatedness across the 1139 genotypes ranged from a minimum of 0.078 to a maximum of 0.263 and a mean of 0.204.

Table 4 presents the two-way contingency table of genotype swap status (`swp`) by genotype packet choice status (`pC`). Only 462 genotypes had sufficient seed to be sown in two plots, while the breeder stipulated that the commercial variety, CBA CAPTAIN, be sown in two plots. The design construction process involves two stages: allocation of packet choice to genotypes and allocation of plots to genotypes, given packet choice status. Stage one determines an optimal subset of size 140 from the genotype swap status "two". In the absence of genetic relatedness this subset would be determined using simple random sampling. For our approach we consider an interim linear mixed model for the pseudo data vector $\boldsymbol{y}$ of length n, which is given by

$$\boldsymbol{y} = \mathbf{1}\mu + \boldsymbol{u_g} + \boldsymbol{\eta} \tag{16}$$

where μ is an overall mean parameter, $\boldsymbol{u_g}$ is the vector of total genetic effects of size $n_{u_g} \times 1$ and $\boldsymbol{\eta}$ is the $n \times 1$ vector of errors. We note that for this interim LMM $n = n_g$. Following Cullis *et al.* (2020) the vector of genetic effects $\boldsymbol{u_g}$ is decomposed into additive and non-additive genetic effects as follows:

$$\boldsymbol{u_g} = \boldsymbol{u_a} + \boldsymbol{u_e} \tag{17}$$

where $\boldsymbol{u_a}$ and $\boldsymbol{u_e}$ are the vectors of additive and non-additive total genetic effects respectively. Substituting (17) into (16) gives

$$\boldsymbol{y} = \mathbf{1}\mu + \boldsymbol{u_a} + \boldsymbol{\eta}^* \tag{18}$$

The random effects $\boldsymbol{u_a}$ and errors $\boldsymbol{\eta}^* = \boldsymbol{u_e} + \boldsymbol{\eta}$ in (18) are assumed normally distributed such that

$$\begin{bmatrix} \boldsymbol{u_a} \\ \boldsymbol{u_e} \\ \boldsymbol{\eta} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \sigma_a^2 \boldsymbol{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_e^2 \boldsymbol{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R} \end{bmatrix} \right)$$

where $\boldsymbol{R} = \oplus_{i=1}^{2} \frac{\sigma^2}{r_i} \boldsymbol{I}_{n_i}$, $n_1 + n_2 = n_g$, $r_1 = 1$ and $r_2 = 2$. Finally, the permute and objective sets are the same and contain the additive genetic effects, and the linked and static sets are *NULL*.

The data frame used for stage one contains variables for genotype, genotype swap status and genotype packet choice status as shown below

```
'data.frame':        1139 obs. of  3 variables:
$ GKeep: Factor w/ 1139 levels "CBACAPTAIN","D16035>F103>19F3TW011",..: 28 46
```

swp	pC1	pC2
capt	0	1
one	677	0
two	321	140

Table 4: Two-way contingency table of genotype swap (`swp`) status (`swp`) by genotype packet choice (`pC`) status for the chickpea S1 experiment.

```
   47 50 51 64 65 75 181 193 ...
$ pC : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 ...
$ swp : Factor w/ 3 levels "capt","one","two": 3 3 3 3 3 3 3 3 3 ...
```

and this data frame, `step1init.df` is then used in the following calls to **odw** to determine an optimal allocation of packet choice to genotypes:

```
require(odw)
sigma.vmg <- 0.8/south.met.abar
sigma.ideg <- 0.2
sigma <- 1
phi <- c(sigma.ideg,sigma)
sv <- odw(fixed=~ 1, random=~ vma(Genotype, south.s1.sweep),
     residual = ~dsum(~units|pC), swap=~swp,
     permute=~ vma(Genotype, south.s1.sweep), start.values = TRUE,
     data=step1init.df)
sv <- sv$vparameters.table
sv$Value <- c(sigma.vmg*south.met.abar,phi[1]+phi[2]/1,
     phi[1]+ phi[2]/2)
temp.od <- odw(fixed=~ 1, random=~ vma(Genotype, south.s1.sweep),
     residual = ~dsum(~units|pC),
     permute=~ vma(Genotype, south.s1.sweep), swap=~swp,
     R.param = sv, G.param = sv, search = 'tabu+rw',
     maxit=20, data=step1init.df)
```

These calls illustrate a number of features found in **odw**. For example, since the variance model is of a specific form, initial values for the variance parameters for the additive genetic effects and the errors must be provided. Use of the argument `start.values` invokes **odw** to return a list containing default variance parameters which we replace with the appropriate values. Variance components for the additive and non-additive genetic effects are set to those observed from fitting similar models to yield data-sets from the Chickpea breeding program. The variance for the errors for each level of `pC` match the variance model for $\boldsymbol{\eta}^*$. Including the non-additive genetic effects with the errors significantly reduces computational time. The `vma` variance model requires specification of a known (scaled) variance matrix, and this is provided in the second argument. In this example, $\boldsymbol{A}^{-1}$ is provided as a sparse matrix held in three column co-ordinate form (as a matrix) in row major order, with the attribute INVERSE set to `TRUE`. This matrix was constructed using the `AIsweep` function found within **pedicure** (see https://mmade.org/pedicure/). The constructor function `dsum` creates the direct sum form for $\boldsymbol{R}$, conditioned on `pC`. The second call to **odw** uses the starting values held in `sv`, in the `G.param` and `R.param` arguments. Lastly, the `swap` argument specifies legal unit exchanges.

The agreement between the initial and **odw** genotype packet status is given by

```
      pCodw
pCinit   1   2
     1 910  88
     2  88  53
```

Stage two of the design construction approach, involves allocation of plots to genotypes. To ensure the final design is resolvable with respect to major blocking components, this is achieved using two steps. The LMM in (2) is used in both steps, and uses the same pseudo data vector of size 1280×1. The permute set are the total genetic effects, in both steps and the variance of these effects is $\boldsymbol{G}_1 = \sigma_a^2 \boldsymbol{A} + \sigma_e^2 \boldsymbol{I}_{n_{u_1}}$. For step one, the static set are the `Trial` and `ColBlock` effects with associated variance matrix $\boldsymbol{G}_2 = \oplus_{i=1}^{2} \sigma_{g_{2_i}}^2 \boldsymbol{I}_{n_{u_{2i}}}$. In step two, the static set are the `Trial, ColBlock, Column` and `Trial:Row` effects with variance $\boldsymbol{G}_2 = \oplus_{i=1}^{4} \sigma_{g_{2_i}}^2 \boldsymbol{I}_{n_{u_{2i}}}$. The `swap` argument is set to `Trial:ColBlock` to ensure that permissable swaps occur between plots within the intersection of `Trial` and `ColBlock`. The error variance matrix for both steps is $\boldsymbol{I}_n$, since $\sigma^2 = 1$. Although the error model used in the analysis of these experiments will be the separable autoregressive process of order one, we have found that designs which are constructed using the separable autoregressive process of order one results in an undesirable allocation of genotypes with a packet status of one. Further, Cullis *et al.* (2020) found that there was little impact on the realised genetic gain by not specifying this variance model for the errors, and hence we use the default variance model for errors. The data-frame containing an initial configuration of plots to genotypes contains the following variables:

```
'data.frame':        1280 obs. of  6 variables:
$ Genotype: Factor w/ 1139 levels "CBACAPTAIN","D16035>F103>19F3TW011",..: 807
   170 524 946 547 901 128 35 724 1024 ...
$ Trial : Factor w/ 2 levels "HS.1","HS.2": 1 1 1 1 1 1 1 1 1 1 ...
$ ColBlock: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
$ Row : Factor w/ 32 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
$ Column : Factor w/ 20 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
$ yield : logi NA NA NA NA NA NA ...
```

The variable `Row` indexes rows within trials, rather than lattice rows, and the major blocking factors `Trial` and `ColBlock` create two blocks in each dimension of the lattice. Figure 1 presents a schematic layout of the experiment which indicates the placement of major blocking factors and the orientation of these with respect to the plot dimensions. The variable `Entry` contains genotype codes. The following four calls to **odw** constructs an optimal allocation of plot to genotypes:

```
# step 2.1
sv <- odw(fixed=~ Site, random=~ vmt(Genotype, south.s1.sweep) +
       Trial + ColBlock, permute=~ vmt(Entry, south.s1.sweep),
       start.values = TRUE, data=init.df)
sv <- sv$vparameters.table
sv$Value[1:2] <- c(sigma.vmg*met.abar,sigma.ideg) # others as defaults
temp.od <- odw(fixed=~ Site, random=~ vmt(Genotype, south.s1.sweep) +
```

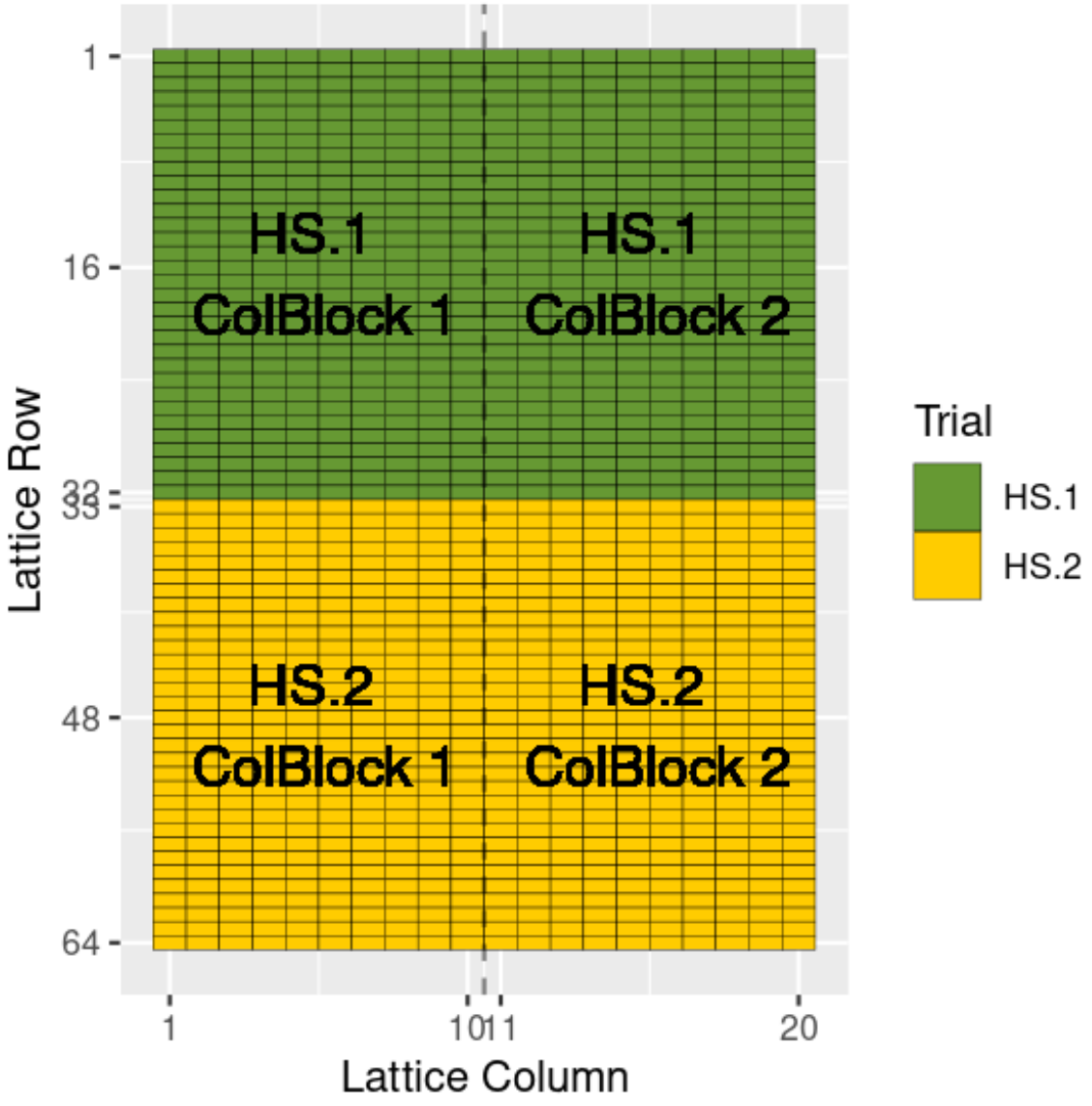

Figure 1: Schematic layout of the S1 chickpea selection experiment conducted at Narrabri in 2022.

```
        Trial + ColBlock,
        residual = ~units, permute=~ vmt(Genotype, south.s1.sweep),
        R.param = sv, G.param = sv, search = 'tabu+rw',
        maxit=20, data=init.df)
# step 2.2
sv <- odw(fixed=~ Site, random=~ vmt(Genotype, south.s1.sweep) +
      Trial + Column + Trial:Row + ColBlock,
      permute=~ vmt(Genotype, south.s1.sweep), swap=~Trial:ColBlock,
      start.values = TRUE, data=temp.od$design)
sv <- sv$vparameters.table
sv$Value[1:2] <- c(sigma.vmg*met.abar,sigma.ideg) # others as defaults
temp.od <- odw(fixed=~ Site, random=~ vmt(Genotype, south.s1.sweep) +
      Trial + Column + Trial:Row + ColBlock,
      residual = ~units, permute=~ vmt(Genotype, south.s1.sweep),
      swap=~Trial:ColBlock,R.param = sv, G.param = sv,
      search = 'tabu+rw',maxit=200, data=temp.od$design)
      summary(temp.od)$is.binary # not for row of course
      xx <- with(temp.od$design,table(Entry,Trial:Row))
      table(xx)
```

The last three commands check the design and returns a list of logical scalars, one for each factor in the model. If `TRUE`, the permute factor is binary with respect to the particular model term. The functionality of `is.binary` within the `summary` method for **odw** does not extend to compound model terms. Hence an additional manual check is made for the `Trial:Row` term.

In order to assess the impact of using genetic relatedness for the construction of designs in terms of their theoretical efficiency we undertook a small study which involved the construction of four design types. These design types are the factorial combinations of using or not using genetic relatedness at each stage of the design construction. Hence the four design types as SG++, SG+-, SG-+ and SG–, where for example SG++ denotes that genetic relatedness was used in both stages of the design construction and SG– denotes that genetic relatedness was not used in either stage. The quality of the design was assessed by computing the $\mathcal{A}$ of the design against the LMM for stage two, step 2 using genetic relatedness. The $\mathcal{A}$ values multiplied by $1e4$, using SG++ as the origin were

```
SG++ SG-+ SG+- SG--
   0   27   15   43
```

These results demonstrate the importance of using genetic relatedness in both stages of the design construction. The impact of using genetic relatedness is more important for stage one than stage two, and the effects appear to be strictly additive.

Designs for additive genetic effects

In the previous section we presented a design construction approach when the aim is to select genotypes from an inbred crop using the prediction accuracy of the vector ot total genetic effects, hence the objective effects were associated with the total genetic effects using the variance model, `vmt`. There are examples where it is of interest to construct as design which is

optimal for additive genetic effects. Some examples include diallel experiments (Butler 2013), selection of parents in hybrid breeding programs, and evaluation of parents using progeny trials in outcrossing plant species (see Cullis, Jefferson, Thompson, and Smith (2014)). These examples would use the vector of additive effects as the objective effects and use `vma` as the variance model in all stages of the design construction process. These ideas can also be used to construct near optimal designs for inbred crops when the number of genotypes is prohibitively large. Cullis *et al.* (2014) used a reduced or an approximate reduced animal model for the analysis of a large MET data-set involving outcrossing plant species and we apply similar ideas to construct a near optimal design for the example presented in the previous section.

The reduced animal model (RAM) is based on the partition of the genetic effects into effects associated with parental and non-parental genotypes. We denote this partition by $\boldsymbol{u_s} = [\boldsymbol{u}_{\boldsymbol{s}_1}^\top \ \boldsymbol{u}_{\boldsymbol{s}_2}^\top]^\top$, $s = g, a$ and e where $\boldsymbol{u_{s_1}}$ and $\boldsymbol{u_{s_2}}$ are the vectors of parental and non-parental genotype effects of size $n_{u_{s1}} \times 1$ and $n_{u_{s2}} \times 1$ respectively. Typically $n_{u_{s1}}$ is much less than n_s. There is a conformal partitioning of the NRM given by

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \tag{19}$$

and we have

$$\boldsymbol{u_{a_2}} = \boldsymbol{T}\boldsymbol{u_{a_1}} + \boldsymbol{u_{m_2}} \tag{20}$$

where $\boldsymbol{u_{m_2}}$ is a vector of mendelian effects for non-parental genotypes of size $n_{u_{a1}} \times 1$, and $\boldsymbol{T}$ is the parent design matrix of size $n_{u_{a1}} \times n_{u_{a2}}$ given by

$$\boldsymbol{T} = \frac{1}{2}(\boldsymbol{F} + \boldsymbol{M})$$

and $\boldsymbol{F}$ and $\boldsymbol{M}$ are female and male design matrices respectively. It follows that

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{11}\boldsymbol{T}^\top \\ \boldsymbol{T}\boldsymbol{A}_{11} & \boldsymbol{T}\boldsymbol{A}_{11}\boldsymbol{T}^\top + \boldsymbol{D}_{22} \end{bmatrix} \tag{21}$$

where $\text{var}(\boldsymbol{u_{m_2}}) = \sigma_a^2 \boldsymbol{D}_{22}$ is a diagonal matrix given by

$$\boldsymbol{D}_{22} = \oplus_{i=1}^{n_{a2}} \left\{ \tfrac{3}{2} - (\tfrac{1}{2})^{s_i} - \tfrac{1}{2}F_{a_i} + ((\tfrac{1}{2})^{s_i} - 1) f_{m_i f_i} \right\}$$

and s_i, F_{a_i}, $f_{m_i f_i}$ are the selfing, average of the inbreeding values for parents and coefficient of coancestry between parents of the ith non-parental genotype respectively. Note that $f_{m_i f_i} = a_{m_i f_i}/2$, where $\boldsymbol{A} = \{a_{ij}\}$. It follows that

$$\boldsymbol{A}^{-1} = \begin{bmatrix} \boldsymbol{A}_{11}^{-1} + \boldsymbol{T}^\top \boldsymbol{D}_{22}^{-1} \boldsymbol{T} & -\boldsymbol{T}^\top \boldsymbol{D}_{22}^{-1} \\ -\boldsymbol{D}_{22}^{-1}\boldsymbol{T} & \boldsymbol{D}_{22}^{-1} \end{bmatrix} \tag{22}$$

This matrix can be obtained using the `ainverse` function in **ODExtras**, which uses the method of Meuwissen and Luo (1992) to generate an inverse relationship matrix in sparse triplet form from a pedigree data frame. The required elements of $\boldsymbol{D}_{22}$ can be extracted as the diagonals of $\boldsymbol{A}^{-1}$ associated with non-parental genotypes in the experiment.

The RAM interim linear mixed model, used for the allocation of packet choice to genotypes is given by

$$\boldsymbol{y} = \boldsymbol{1}\mu + \boldsymbol{Z_g}\boldsymbol{u_g} + \boldsymbol{\eta} \tag{23}$$

where $\boldsymbol{Z_g} = [\boldsymbol{Z}_{g_1}\ \boldsymbol{Z}_{g_2}]$ is the conformal partitioning of $\boldsymbol{Z_g}$ associated with parental and non-parental genotypes effects in $\boldsymbol{u_g}$. Substituting (17) and (20) into (23) gives

$$\boldsymbol{y} = \mathbf{1}\mu + (\boldsymbol{Z}_{g_1} + \boldsymbol{Z}_{g_2}\boldsymbol{T})\boldsymbol{u}_{a_1} + \boldsymbol{Z}_{g_2}\boldsymbol{u}_{m_2} + \boldsymbol{Z}_{g_1}\boldsymbol{u}_{e_1} + \boldsymbol{Z}_{g_2}\boldsymbol{u}_{e_2} + \boldsymbol{\eta} \tag{24}$$

In many cases there is no data on parental genotypes, particularly in selection experiments in hybrid crops and outcrossing species, so $\boldsymbol{Z}_{g_1} = \mathbf{0}$. However for complete generality we retain terms associated with $\boldsymbol{Z}_{g_1}$ in the model. The design matrix for $\boldsymbol{u}_{a_1}$ is non-standard, but this can be accommodated in **odw** by using the `xpr` constructor function. Further simplifications for (24) can be achieved by considering the partition of $\boldsymbol{u}_{s_1}$ given by $\boldsymbol{u}_{s_1} = [\boldsymbol{u}_{s_{11}}^\top \boldsymbol{u}_{s_{12}}^\top]^\top, s = g, a$ and e where $\boldsymbol{u}_{s_{11}}$ and $\boldsymbol{u}_{s_{12}}$ are the vectors of effects for parental genotypes without and with data of size $n_{u_{s11}} \times 1$ and $n_{u_{s12}} \times 1$ respectively. Hence, assuming all non-parental genotypes have data then $n_{u_{s12}} + n_{u_{s2}} = n$. Using the partition of $\boldsymbol{Z}_{g_1} = [\boldsymbol{Z}_{g_{11}}\ \boldsymbol{Z}_{g_{12}}]$ in (24) leads to

$$\boldsymbol{y} = \mathbf{1}\mu + (\boldsymbol{Z}_{g_1} + \boldsymbol{Z}_{g_2}\boldsymbol{T})\boldsymbol{u}_{a_1} + \boldsymbol{Z}_{g_2}\boldsymbol{u}_{m_2} + \boldsymbol{\eta}^* \tag{25}$$

where $\boldsymbol{\eta}^* = \boldsymbol{u}_e^* + \boldsymbol{\eta}$ and $\boldsymbol{u}_e^* = [\boldsymbol{u}_{e_{12}}^\top \boldsymbol{u}_{e_2}^\top]^\top$. This simplified formulation holds as

$$[\boldsymbol{Z}_{g_{12}}\ \boldsymbol{Z}_{g_2}][\boldsymbol{Z}_{g_{12}}\ \boldsymbol{Z}_{g_2}]^\top = \boldsymbol{I}_n$$

for all permutations applied to the rows of $[\boldsymbol{Z}_{g_{11}}\ \boldsymbol{Z}_{g_{12}}]$.

Equation (25 is still not suitable to reduce the computational time relative to using the model in (18. This is due to the presence of the Mendelian sampling term, which has a large number of effects. Although the size of the objective set is $n_{u_{s11}}$, the non-parental Mendelian effects are in the linked set. A useful approximation which overcomes this problem is to replace $\boldsymbol{D}_2 2$ by $\bar{d}_{22}$. Use of this approximation means that $\boldsymbol{u_{m_2}}$ can be incorporated into the errors, and this is demonstrated in the following. There are 43 parents for the 1139 genotypes in the S1 example. Of these three are also present in the experiment. These genotypes are CBA CAPTAIN, PBA STRIKER and PBA SLASHER, but only CBA CAPTAIN occurs as a parent for the non-parental genotypes. The inbreeding coefficients for these parental genotypes are $0.9854, 0.9961$ and 0.9961, hence it is plausible to replace the parents for these genotypes in the initial data frame by themselves. This results in a total of 43 parental genotypes, but we note that the appropriate parents are retained for these genotypes in forming the NRM for the parental genotypes.

As for the full approach, initially factors for the genotype swap status and genotype packet choice status are created. Table 5 presents the two-way contingency table of genotype swap status (`swp`) by genotype packet choice status (`pC`). The levels of `pC` correspond to the conditional variance of the data, in conjuction with the swap block status:

$$\mathrm{var}(\boldsymbol{y}|\boldsymbol{u}_{a_1}) = \begin{cases} \sigma_e^2 + \sigma^2/2 & : \text{ pC1 - CBA CAPTAIN, two packets} \\ \sigma_e^2 + \sigma^2 & : \text{ pC2 - PBA SLASHER/PBA STRIKER, one packet} \\ \sigma_e^2 + \sigma^2/2 & : \text{ pC3 - PBASLASHER/PBASTRIKER, two packets} \\ \bar{d}_{22}\sigma_a^2 + \sigma_e^2 + \sigma^2 & : \text{ pC4 - non parental genotypes, one packet} \\ \bar{d}_{22}\sigma_a^2 + \sigma_e^2 + \sigma^2/2 & : \text{ pC5 - non parental genotypes, two packets} \end{cases}$$

The conditional variance for non parental genotypes includes $\bar{d}_{22}\sigma_a^2$ from the Mendelian sampling effect. Note that the breeder requires that CBA CAPTAIN be included with two packets, and at least one of the other check genotypes be included with two packets. Only 459 non

parental genotypes had sufficient seed to be sown in two plots. We note that levels `pC1` and `pC3` of `pC` could be merged in this example, but for pedagogical reasons we retain these as separate levels to make the distinction between CBA CAPTAIN and the other two parental genotypes clear. As before, the design construction process involves two stages: stage one - allocation of packet choice to genotypes and, stage two - allocation of plots to genotypes, given packet choice status.

swp	pC1	pC2	pC3	pC4	pC5
capt	1	0	0	0	0
ch(slash/str)	0	1	1	0	0
one	0	0	0	677	0
two	0	0	0	320	139

Table 5: Two-way contingency table of genotype swap (`swp`) status (`swp`) by genotype packet choice (`pC`) status for the RAM model with the chickpea S1 experiment.

The data frame used for stage one contains variables for genotype, Female parent, Male parent, genotype swap status and genotype packet choice status as shown below

```
'data.frame':        1139 obs. of  5 variables:
$ GKeep : Factor w/ 1139 levels "CBACAPTAIN","D16035>F103>19F3TW011",..: 1139 1
   1138 2 3 4 5 6 7 8 ...
$ Female: Factor w/ 37 levels "05109-1279/CICA1841",..: 37 7 36 10 10 10 19 19
   19 19 ...
$ Male : Factor w/ 15 levels "CBA2042","CBACAPTAIN",..: 15 2 14 6 6 6 3 3 3 3
   ...
$ swp : Factor w/ 4 levels "capt","ch(slash/str)",..: 1 2 2 3 4 4 3 3 3 3 ...
$ pC : Factor w/ 5 levels "1","2","3","4",..: 1 2 3 4 4 4 4 4 4 4 ...
```

and this data frame, `step1ram.init.df` is then used as the initial configuration for **odw** to determine an optimal allocation of packet choice to genotypes:

```
sigma.vmg <- 0.8/real.south.met.parabar
sigma.ideg <- 0.2
sigma.mend <- dbar22*sigma.vmg
sigma <- 1
phi <- c(sigma.mend,sigma.ideg,sigma)

sv <- odw(fixed=~ 1,
        random=~ str(~xpr(~Female/2+Male/2),~vma(Female,real.s1.par.sweep)),
        residual = ~dsum(~id(units)|pC),
        swap=~swp,optimize = real.parents.in.south,
        permute=~ xpr(~Female/2+Male/2),
        equate.levels=c('Female','Male'),
        start.values = TRUE, data=step1ram.init.df,
        reorder = c('GKeep'))
sv <- sv$vparameters.table
sv$Value <- c(sigma.vmg,
```

```
sigma.ideg + sigma/2,
sigma.ideg + sigma,sigma.ideg + sigma/2,
sigma.vmg * dbar22 + sigma.ideg + sigma,
sigma.vmg * dbar22 + sigma.ideg + sigma/2)
sv
temp.od <- odw(fixed=~ 1,
        random=~ str(~xpr(~Female/2+Male/2),~vma(Female,real.s1.par.sweep)),
        residual = ~dsum(~id(units)|pC),
        swap =~swp,optimize = real.parents.in.south,
        permute=~ xpr(~Female/2+Male/2),
        equate.levels = c('Female','Male'),
        R.param = sv, G.param = sv, data=step1ram.init.df,
        reorder = c('GKeep'),
        maxit=4, search='tabu+rw')
temp.od <- update(temp.od,maxit=60)
```

These calls introduce some additional features of **odw**. For example, the non standard design matrix for $\boldsymbol{u}_{a_1}$ is formed using `xpr`, the optimality criteria be computed using `optimize` levels only, `equate.levels=c('Female','Male')` specifies that the levels of `Female` and `Male` match and `reorder = c('GKeep')` permutes `GKeep` (at termination of the search) in design order, parallel to `xpr(~Female/2+Male/2)`.

The agreement between the initial and **odw** genotype packet status is given by

```
      pCodw
pCinit   1   2   3   4   5
     1   1   0   0   0   0
     2   0   1   0   0   0
     3   0   0   1   0   0
     4   0   0   0 882 115
     5   0   0   0 115  24
```

The calls for stage two are very similar to those when using the full model. An example of the call for step 2 in stage 2 is below

```
sv <- odw(fixed=~ 1,
        random=~ str(~xpr(~Female/2+Male/2),~vm(Female,real.s1.par.sweep)) +
        Trial + Column + Trial:Row + ColBlock,
        residual = ~units, swap=~Trial:ColBlock,optimize = real.parents.in.south,
        permute=~ xpr(~Female/2+Male/2), equate.levels=c('Female','Male'),
        start.values = TRUE, data=temp.od$design, reorder = c('GKeep'))
sv <- sv$vparameters.table
sv$Value[1] <- c(sigma.vmg)
sv
temp.od <- odw(fixed=~ 1,
        random=~ str(~xpr(~Female/2+Male/2),~vm(Female,real.s1.par.sweep)) +
        Trial + Column + Trial:Row + ColBlock,
        residual = ~units, swap=~Trial:ColBlock,optimize = real.parents.in.south,
```

```
        permute=~ xpr(~Female/2+Male/2), equate.levels=c('Female','Male'),
        R.param = sv, G.param = sv, data=temp.od$design, reorder = c('GKeep'),
        maxit=4, search='tabu+rw')
temp.od <- update(temp.od,maxit=500)
```

Note that it was not possible to achieve resolvability with respect to the major blocking factors for the RAM approach. The quality of the RAM design was assessed by computing the $\mathcal{A}$ against the full LMM for stage two, step 2 using genetic relatedness. The $\mathcal{A}$ values multiplied by $1e4$, using SG++ as the origin were

```
SG++ SG-+ SG+- SG--  RAM
   0   27   15   43    9
```

These results suggest that there is little loss in efficiency using the RAM approach. This is a very useful result since the reduction in computing time is substantial. For example, on a Dell Inc. Precision 7560, with 128.0 GiB RAM, an Intel®Xeon(R) W-11955M CPU @ 2.60GHz × 16 processor and 2.0 TB disk space running Ubuntu 22.04.1 LTS, there was a 50 fold reduction in time per tabu loop for the RAM model for the stage two, step two call to **odw** from 20.25 seconds to 0.4 seconds.

Determining optimal subsets for selective phenotyping

Huang, Clifford, and Cavanagh (2013) consider a method to maximize genetic diversity within the selected samples for so-called selective phenotyping. Here we propose an alternate approach using **odw**, which can be used in any design, and with either ancestral or marker based relationship matrices. Our approach is a simple extension to the allocation of packet choice status to genotypes. To illustrate the approach we again consider the S1 chickpea experiment, but for this application, assume that there is only sufficient resources to sow 640 plots. Hence the aim is to determine an optimal subset to not phenotype, but maintain approximately 10% partial replication, while accounting for seed supply constraints and breeder choices.

As for the earlier models, factors for the genotype swap status and genotype packet choice status have to be created. Table 5 presents the two-way contingency table of genotype swap status (`swp`) by genotype packet choice status (`pC`). The levels of `pC` correspond to the conditional variance of the data and these provide **odw** with the to find an optimal allocation with respect to additive genetic effects:

$$\text{var}(\boldsymbol{y}|\boldsymbol{u}_{\boldsymbol{a}_1}) = \begin{cases} \sigma_e^2 + \sigma^2 & : \text{ pC1 - one packet} \\ \sigma_e^2 + \sigma^2/2 & : \text{ pC2 - two packets} \\ \sigma_e^2 + \alpha\sigma^2 & : \text{ pC3 - no packets} \end{cases}$$

The conditional variance takes three values, where α is set to a large number, say 300. This idea is based on the so-called alternative outlier model of Thompson (1985), where data points to be excluded can be regarded as those associated with an inflated error variance. This makes the approach simple to implement in **odw** and intuitively appealing. The genotypes swap status was similar to before so that the breeder requires that CBA CAPTAIN be included with two packets, and at least one of the other check genotypes be included with two packets. A total of 568 genotypes would not be phenotyped. As before, the design construction process involves two stages: stage one - allocation of packet choice to genotypes and, stage two - allocation of plots to genotypes, given packet choice status.

swp	pC1	pC2	pC3
capt	0	1	0
ch(slash/str)	1	1	0
one	338	0	339
two	163	67	229

Table 6: Two-way contingency table of genotype swap (`swp`) status (`swp`) by genotype packet choice (`pC`) status for selective phenotyping with the chickpea S1 experiment.

The code to determine the allocation of packet choice status to genotypes in **odw** is:

```
sigma.vmg <- 0.8/south.met.abar
sigma.ideg <- 0.2
sigma <- 1
alpha <- 300
phi <- c(sigma.ideg,sigma,alpha*sigma)
sv <- odw(fixed=~ 1, random=~ vm(GKeep, south.s1.sweep),
        residual = ~dsum(~units|pC), swap=~swp,
        permute=~ vm(GKeep, south.s1.sweep), start.values = TRUE,
        data=step1drop.init.df)
sv <- sv$vparameters.table
sv$Value <- c(sigma.vmg*south.met.abar,phi[1]+phi[2]/1,
        phi[1]+ phi[2]/2, phi[1]+phi[3])
sv
temp.od <- odw(fixed=~ 1, random=~ vm(GKeep, south.s1.sweep),
        residual = ~dsum(~units|pC),
        permute=~ vm(GKeep, south.s1.sweep), swap=~swp,
        R.param = sv, G.param = sv, search = 'tabu+rw',
        maxit=20, data=step1drop.init.df)
```

The agreement between the initial and **odw** genotype packet status is given by

```
      pCodw
pCinit   1   2   3
     1 238  22 242
     2  19  19  31
     3 245  28 295
```

The method of Huang *et al.* (2013) was also used to construct a subset of test lines to phenotype. Using this approach the 1136 test lines were subsetted and then combined with the three checks. Out of a total of 571 genotypes there were only 363 in common between this subset and the subset selected using **odw**. Using the Huang *et al.* (2013) approach still requires determining which genotypes will be replicated from the subset of phenotyped genotypes. This is clearly piecemeal and inefficient, whereas the approach using **odw** finds both subsets in a single step, respecting genetic relatedness between test lines and check lines.

5.4. Multi-phase design

In this example we consider the design of an experiment to predict the performance of wheat varieties in terms of the important quality trait of falling number (FN). The measurement of FN requires a multi-phase experiment (Brien 1983; Smith, Lim, and Cullis 2006) with two phases. The varieties are first grown in a field experiment (Phase I) and after this has been harvested, grain samples from individual plots are processed in a laboratory experiment (Phase II) to obtain the trait of interest. Several authors, including Brien (1983) and Smith *et al.* (2006), have stressed the need for the use of valid experimental designs for all phases of a multi-phase experiment. However, this rarely occurs in practice and one of the major impediments has been the lack of suitable software to generate the designs. In this example we illustrate how **odw** can produce valid and efficient designs for the seemingly complex scenario of multi-phase experiments.

The Phase I field experiment in this example comprises 144 plots arranged in a rectangular array of 24 columns by 6 rows. A total of 105 varieties will be grown using a partially replicated ($p-$rep) design (Cullis, Smith, and Coombes 2006b) in which 39 varieties will be planted in two plots each while the remaining 66 varieties will be planted in single plots. Unlike previous examples in this section, there is no information available on the genetic relatedness of varieties so that replicated varieties will be chosen at random.

The laboratory experiment in Phase II involves the production of a slurry from each grain sample taken from the field. The slurry is then placed in a tube on one of the two available FN machines to measure the trait, which is the time taken (in seconds) for a rod to travel through the slurry. In this experiment, samples from all 144 field plots will be processed, and replication in Phase II will be achieved by producing two slurries (from two separate grain samples) for a subset of the plots. A partially replicated ($q-$rep) design (Smith *et al.* 2006) will be used in which 40 plots will be tested using two slurries while the remaining 104 plots will be tested as single slurries, making a total of 184 slurries to be processed. The choice of plots to be replicated in Phase II can be made in an informed manner, and this will be discussed in Section 5.7. The slurries will be processed using two FN machines, each of which comprises two tubes. This allows four slurries to be processed simultaneously, and these will be referred to as a run. Thus the full Phase II design will require 46 runs which are processed sequentially. Practical considerations necessitate the grouping of runs into blocks with (no more than) 8 runs in each, and with 3 blocks per day. The full design spans 2 days, and the final block on each day will have 7 rather than 8 runs. The schematic layout of the Phase II design is shown in Figure 2. Note that this also contains information relating to the final design which will be discussed in section 5.6.

The design of multi-phase experiments can be achieved by considering each phase in turn, with the design for higher order phases being conditional on the preceding phases. In terms of model based design, this hierarchical structure adds some complexities to the LMMs required for individual phases. In this example we therefore use the Design Tableau (DT) approach of Smith and Cullis (2018), which was developed to aid with the specification of a LMM for analysis, given an experimental design. In the DT approach, the key components that must be defined are the factors associated with the treatments, the factors associated with the observational units (also generically called the "plots") and the design function. In the standard application of DT, the design function relates to the manner in which treatments were allocated to plots. In our application it relates to the design search we are about to undertake. With this base-line information we can derive the so-called treatment and plot structures, each of which is expressed as a model formula using the notation of Wilkinson and

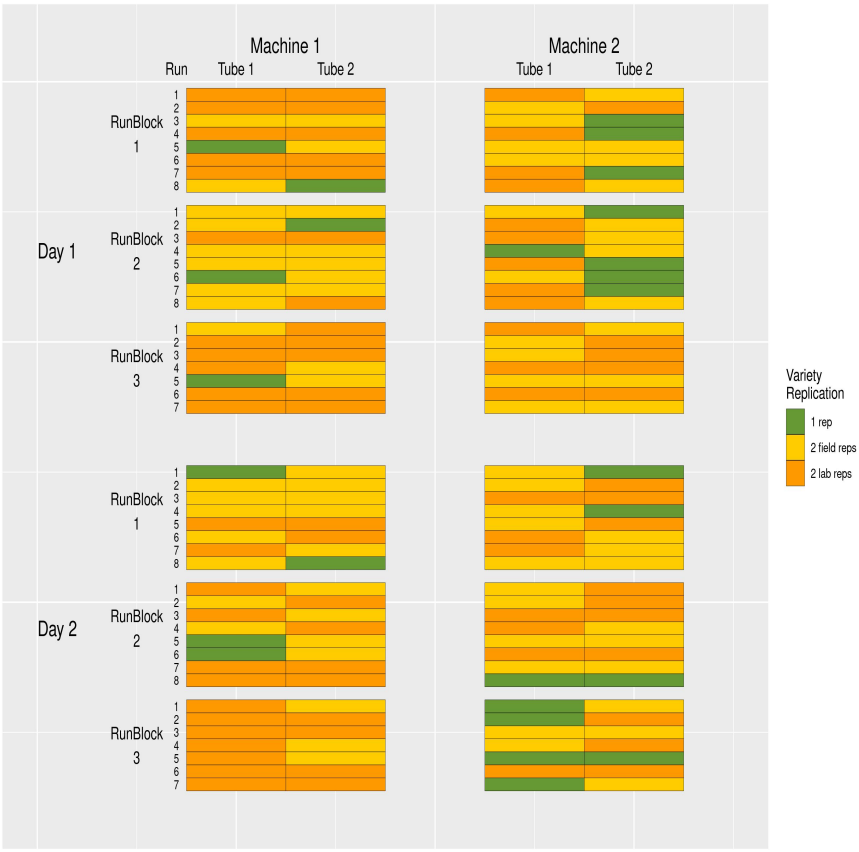

Figure 2: Schematic layout of second phase for multi-phase experiment example.

Rogers (1973b). In the context of **odw**, the treatment formula is associated with the `permute` formula, and the combination of the treatment and plot formulae provide the overall form of the LMM. In sections 5.5 and 5.6 we summarise the derivation of the **odw** calls for the example using the DT procedure. The reader is referred to Smith and Cullis (2018) for full details of the Design Tableau approach.

5.5. Phase I design

In order to construct the design for the first phase we ignore the fact that there are further phases so assume a trait is to be measured directly from the field experiment. The key steps in the DT process for construction of the Phase I design are as follows:

1. Define the plots (observational units) and the plot factors: the observational units are the field plots and there are 144 of these. The plot factors, with the number of levels given in parentheses are: `ColBlock` (2), `Column` (24) and `Row` (6), where `ColBlock` corresponds to blocks of contiguous columns (columns 1-12 and 13-24 for `ColBlock` 1 and 2 respectively). Note that it must be possible to uniquely index the observational units using the full set of plot factors. In this example they are completely indexed using `Column` and `Row`
2. Define the treatments and treatment factors: the treatments are the varieties. The treatment factors are: `Variety` (105)
3. Describe the design function/search: the search will aim for resolvability (aligned with column blocks) for the varieties with multiple observational units and will allow for sources of variation associated with rows and columns; the objective will be $\mathcal{A}-$optimality, to minimise the average pairwise prediction error variance between the varieties
4. Define the treatment model formula: `Variety`
5. Define the plot model formula (as implied by the design function): `ColBlock + Column*Row = ColBlock + Column + Row + Column:Row`.

In terms of the **odw** call, the *permute* set, are the set of variety effects. Since this contains a single term only, the objective effects are also the variety effects. The `fixed`, `random` and `residual` formulae are extracted from the combined treatment and plot model formulae. The terms in the latter are always included as random effects but note that the final term, namely `Column:Row`, indexes the observational units so defines the error term. It is therefore implicitly included in the **odw** call as the `residual` formula. In the current example the only fixed effect is an overall mean so that the `Variety` term in the treatment model is included as a random term. The **odw** call to construct this design is therefore given by:

```
PhaseI.od <- odw(fixed=~ 1, random=~ Variety + ColBlock + Column + Row,
residual = ~units, permute=~ Variety,
search = 'tabu+rw',maxit=10, data=init.df)
PhaseI.od <- update(PhaseI.od,maxit=10)
```

where `init.df` is the data-frame containing the initial design and contains 144 records, indexed by field plots. Note that in this example, the requirement for resolvable blocks was met using the single **odw** call.

5.6. Phase II design

The key steps in the DT process for construction of the Phase II design are as follows:

1. Define the plots (observational units) and the plot factors: the observational units are the slurries and there are 184 of these. The plot factors, with the number of levels given in parentheses are: `Day` (2), `RunBlock` (3), `Run` (8), `Machine` (2) and `Tube` (2), where `RunBlock` corresponds to blocks of consecutive runs within each day. The observational units are completely indexed using all five factors.

2. Define the treatments and treatment factors: here we must invoke the full definition of treatments as the "entire description of what can be applied to an experimental unit" (Bailey 2008a). Thus in the second phase, the treatment factors comprise not only the varieties, but the complete set of factors associated with the field plots in which the varieties were grown. The treatment factors are therefore: `Variety` (105), `ColBlock` (2), `Column` (24) and `Row` (6). For notational convenience, we also define the factor `FieldPlot` (144) which can be used instead of `Column:Row` in model formulae.

3. Describe the design function/search: the search will aim for resolvability with respect to two factors, namely days and machines, for the varieties with multiple observational units (i.e. slurries). It will allow for sources of variation associated with both the temporal aspect of the measurement process and the natural physical sources associated with machines and tubes within machines; the objective effects are the `Variety` effects and as before we use $\mathcal{A}-$optimality

4. Define the treatment model formula: the treatment formula for the second phase is given by the combined treatment and plot formulae from the first phase (Smith and Cullis 2018): `Variety + ColBlock + Column + Row + Column:Row`

5. Define the plot model formula (as implied by the design function): `(Day/RunBlock/Run) * (Machine/Tube)`

Once again the `permute` formula in **odw** is given by the treatment model formula. However the latter now includes multiple terms and these align with the partitioning of the `permute` formula into objective and linked sets of terms. Importantly, the objective term, appearing before the "|" symbol is given by `Variety` so that the optimality criterion relates to the aims of the experiment, namely the prediction of variety effects. The linked terms, that is those terms after the "|", comprise all other terms from Phase I, since the associated factors must be permuted with `Variety` in the design search but should not contribute to the optimality criteria. Unlike the first phase, the requirement for resolvable blocks in Phase II was not achievable using a single **odw** call so we used the two-step approach. The **odw** call for the first step is given by:

```
PhaseIIbin.od <- odw(fixed=~ 1, random=~Variety + Day + Machine,
residual = ~units, permute=~ Variety,
reorder = c('FieldPlot','Column','Row','ColBlock'),
search = 'tabu+rw', maxit=10, data=init.lab.df)
PhaseIIbin.od <- update(PhaseIIbin.od, maxit=10)
```

where `init.lab.df` is the data-frame containing the initial design (based on the laboratory replication of field plots as determined in Section 5.7) and contains 184 records, indexed by slurries.

The resultant design (resolvable for two factors) was then used as the starting design for the final search. Three designs were created, each using a different model for the `swap` formula, in order to investigate the impact on $\mathcal{A}-$optimality of the restrictions imposed by resolvability. The formulae were: (1) `swap=~Day:Machine`, (2) `swap=~Day` and (3) `swap=~NULL`. The first formula restricts permutations to within the intersection of `Day` and `Machine`; the second to within `Day` and the third has no restrictions. Apart from the `swap` formula, the **odw** call to construct all three designs was identical. The call for the first design is given by:

```
PhaseII.od <- odw(fixed=~ 1, random=~ Variety + FieldPlot +
    Column + Row + ColBlock + Day +
    Day:RunBlock+Day:RunBlock:Run + Machine + Machine:Tube +
    Day:Machine + Day:RunBlock:Machine + Day:RunBlock:Run:Machine +
    Day:Machine:Tube + Day:RunBlock:Machine:Tube, swap=~Day:Machine,
    residual = ~units, permute=~ Variety|FieldPlot + Column + Row + ColBlock,
    search = 'tabu+rw', maxit=10, data=PhaseIIbin.od$design)
PhaseII.od <- update(PhaseII.od, maxit=50)
```

This design call demonstrates the use of the"|" operator which separates the objective and linked terms within the `permute` formula. The $\mathcal{A}-$values for the three designs were (1) 0.179415, (2) 0.179284 and (3) 0.179202. Thus there was a penalty associated with the restrictions for resolvability. Note that in the second design, the removal of the resolvability restriction associated with `Machine` led to a non-binary design for this factor. Similarly, the third design, with the lowest $\mathcal{A}-$value, was non-binary for both *Day* and *Machine*. This was regarded as undesirable by the researcher, so, as a compromise, the design with resolvability for days alone was adopted.

The ability to permute varieties together with linked factors and to optimise on variety effects is a unique and key feature of **odw** for the design of second (and higher order phases. Previously, in the second phase design, it was only possible to permute field plots and optimise on field plot effects. In order to assess the impact of using `Variety` as the objective term rather than `FieldPlot`, a small study which involved the construction of four design types was conducted. The design types comprised the factorial combinations of using or not using `Variety` as the objective term and the two stages of the Phase II design construction, namely the choice of field plots to replicate (see Section 5.7) and the final allocation of varieties as described above. The designs are labelled P2D++ (`Variety` as objective in both stages), P2D+−(`Variety` as objective in first stage only), P2D−+ (`Variety` as objective in second stage only) and P2D−− (`Variety` not used as objective in either stage). Thus P2D++ corresponds to design (2) as described above. The quality of each design was assessed by computing the $\mathcal{A}-$value of the design against the LMM used for P2D++ and is reported here as the difference from P2D++ multiplied by $1e6$. The resultant differences were 0 for P2D++; 622 for P2D+−; 58 for P2D−+ and 675 for P2D−−. Thus the loss associated with not using `Variety` as the objective function was substantially larger for stage 2 (the final allocation in the laboratory).

5.7. Selecting field plots to replicate in Phase II

In order to determine an optimal subset of 40 field plots from Phase I to be replicated in Phase II, we consider an interim LMM for Phase II in which the pseudo data vector represents field plot means (across laboratory replicates) rather than observations for individual slurries. Thus we write

$$\boldsymbol{y} = \mathbf{1}\mu + : \boldsymbol{Z_g} : \boldsymbol{u_g} + : \boldsymbol{Z_c} : \boldsymbol{u_c} + : \boldsymbol{Z_r} : \boldsymbol{u_r} + : \boldsymbol{Z_b} : \boldsymbol{u_b} + : \boldsymbol{u_p} + : \boldsymbol{\eta}$$

where : $\boldsymbol{y}$ is the pseudo data vector of length $n = 144$, μ is an overall mean parameter, : $\boldsymbol{u_g}$, : $\boldsymbol{u_c}$, : $\boldsymbol{u_r}$ and : $\boldsymbol{u_b}$ are the vectors of random `Variety`, `Column`, `Row` and `ColBlock` effects. The last two vectors, : $\boldsymbol{u_p}$ and : $\boldsymbol{\eta}$, both have length n and represent the field plot effects and Phase II errors, respectively. The random effects are assumed to be normally distributed and independent (both between and within sets) with variance components σ_g^2, σ_c^2, σ_r^2, σ_b^2 and σ_p^2. We assume that the pseudo data vector is ordered such that the first 104 "observational units" correspond to field plot means based on a single slurry and the remaining 40 to means based on two slurries. We then assume the variance matrix for : $\boldsymbol{\eta}$ is given by $\oplus_{i=1}^{2} \frac{\sigma^2}{r_i} : \boldsymbol{I}_{n_i}$ where $n_1 = 104,\ n_2 = 40,\ r_1 = 1,\ r_2 = 2$ and σ^2 is the Phase II error variance component. It is computationally efficient to then define a combined vector of errors, namely : $\boldsymbol{\eta}^* =: \boldsymbol{u_p} + : \boldsymbol{\eta}$ so that var : $\boldsymbol{\eta}^* = \oplus_{i=1}^{2} \left(\sigma_p^2 + \frac{\sigma^2}{r_i}\right) : \boldsymbol{I}_{n_i}$. In order to define this heterogeneous error variance structure in **odw** we require the initial data-frame to include a two level factor (called `pC` to be consistent with the other examples), which has the value 1 for the first 104 records and 2 for the remainder.

We then use **odw** to allocate levels of laboratory replication (associated with the factor `pC`) to varieties as follows:

```
PhaseIIrep.od <- odw(fixed=~ 1, random=~ Variety + Column + Row + ColBlock,
residual = ~dsum(~units|pC),
permute=~ Variety|ColBlock + Column + Row,
R.param = sv, G.param = sv, search = 'tabu+rw', data=dup.df,
reorder = c('FieldPlot'),maxit=10)
```

where `dup.df` is the data-frame containing the initial design and contains 144 records, indexed by field plots. Note that the variance parameter values used to form the starting values in `sv` were $\sigma_g^2 = 1.0,\ \sigma_c^2 = 0.1,\ \sigma_r^2 = 0.1,\ \sigma_b^2 = 0.1,\ \sigma_p^2 = 0.5$ and $\sigma^2 = 1.0$. Thus the last two values in `sv` which relate to the combined vector of errors were 1.5 and 1.0 for levels 1 and 2 of `pC`, respectively.

A key feature of the allocation of laboratory replication from this **odw** call is that the field plots to be replicated in Phase II corresponded only to varieties that were *not* replicated in Phase I:

	pC=1	pC=2
Varieties with 1 field plot	26	40
Varieties with 2 field plots	39	0

This allocation formed the basis of the initial design used for Phase II (see section 5.6). Note that it means there were 79 varieties with 2 slurries (39 varieties with a single slurry from

each of 2 field plots and 40 varieties with 2 slurries from a single field plot) and 26 varieties with a single slurry, making a total of 184 slurries, as required.

6. Summary and discussion

This paper presents a formal model based approach to the design of comparative experiments implemented in the R package **odw**. The focus is on the functionality allowed by this paradigm, coupled with an efficient computing strategy for optimal designs with qualitative treatments, where both treatment effects and residuals are correlated. Though not exhaustive, the examples are intended as a guide to several operating characteristics of the package that we have found useful in addressing certain design challenges encountered in practice. The examples are not prescriptive; the models and parameter values chosen are not meant to imply that they are suitable in other settings.

A strength of the package is an easy to use model based interface using R formulae objects that strongly links the design and analysis chain. Helper functions assist in setting random parameters, and `plot` and `summary` methods provide visual and tabular diagnostics, respectively. Also, design specification in **odw** is data driven; the supplied data frame encompasses the physical or temporal blueprint, and does not enforce any configuration restrictions, thus making possible differing block sizes or unusual blocking arrangements. A caveat to this advantage arises from the assumption of separability at the residual level, the consequence being that for correlated error structures a non-rectangular layout must be accommodated by the judicious use of `swap` and additional dummy treatments to complete the array.

At present the data frame expected by **odw** must be constructed by the user outside the package environment. This may be the only recourse for complex designs, however, add-ons to the package could include concise but intuitive methods to construct initial data frames for less demanding configurations or standard design types. A significant avenue for improvement in practice is the integration of the predict formulation outlined in Section 3.2.

References

Bailey R (2008a). *Design of Comparative Experiments*. Cambridge University Press, Cambridge.

Bailey RA (2008b). *Design of Comparative Experiments*. Cambridge University Press.

Brien CJ (1983). "Analysis of Variance Tables Based on Experimental Structure." *Biometrics*, **39**, 53–59.

Bueno Filho JSS, Gilmour SG (2003). "Planning Incomplete Block Experiments When Treatments are Genetically Related." *Biometrics*, **59**, 375–381.

Bueno Filho JSS, Gilmour SG (2007). "Block Designs for Random Treatment Effects." *Journal of Statistical Planning and Inference*, **137**, 1446–1451.

Butler DG (2013). *On the Optimal Design of Experiments Under the Linear Mixed Model*. Ph.D. thesis, School of Mathematics and Physics, The University of Queensland.

Butler DG, Eccleston JA, Cullis BR (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.

Chan BSP (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.

Coombes NE (2009). "DiGGeR, a Spatial Design Program." *Biometric Bulletin*, NSW DPI.

Cullis BR, Jefferson P, Thompson R, Smith AB (2014). "Factor Analytic and Reduced Animal Models for the Investigation of Additive Genotype-by-Environment Interaction in Outcrossing Plant Species with Application to a Pinus Radiata Breeding Programme." *Theoretical and Applied Genetics*, **127**(10), 2193–2210. ISSN 0040-5752. doi:10.1007/s00122-014-2373-0.

Cullis BR, Lill W, Fisher J, Read B, Gleeson A (1989). "A New Procedure for the Analysis of Early Generation Variety Trials." *Applied Statistics*, **38**(2), 361–375.

Cullis BR, Smith AB, Cocks NA, Butler DG (2020). "The Design of Early-Stage Plant Breeding Trials Using Genetic Relatedness." *Journal of Agricultural, Biological and Environmental Statistics*, **25**(4), 553–578. doi:10.1007/s13253-020-00403-5. URL https://doi.org/10.1007/s13253-020-00403-5.

Cullis BR, Smith AB, Coombes NE (2006a). "On the Design of Early Generation Variety Trials with Correlated Data." *Journal of Agricultural, Biological, and Environmental Statistics*, **11**, 381–393.

Cullis BR, Smith AB, Coombes NE (2006b). "On the Design of Early Generation Variety Trials with Correlated Data." *Journal of agricultural, biological, and environmental statistics*, **11**(4), 381–393.

Eccleston JA, Whitaker D (1999). "On the Design of Optimal Change-Over Experiments Through Multi-Objective Simulated Annealing." *Statistics and Computing*, **9**, 37–42.

Elliot LJ, Eccleston JA, Martin RJ (1999). "An Algorithm for the Design of Factorial Experiments When the Data Are Correlated." *Statistics and Computing*, **9**, 195–201.

Fowler G, Knoll L, Vo K, Eastlake D (2011). "The FNV Non-Cryptographic Hash Algorithm, draft-eastlake-fnv-02.txt, Work in Progress." URL http://www.isthe.com/chongo/tech/comp/fnv/index.html.

Gilmour A, Cullis B, Welham S, Gogel B, Thompson R (2004). "An Efficient Computing Strategy for Prediction in Mixed Linear Models." *Computational Statistics and Data Analysis*, **44**, 571–586.

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.

Harville DA (1997). *Matrix Algebra from a Statistician's Perspective*. Springer-Verlag, New York.

Huang E, Clifford D, Cavanagh C (2013). "Selecting Subsets of Genotyped Experimental Populations for Phenotyping to Maximize Genetic Diversity." *Theoretical and Applied Genetics*, **126**(2), 379–88. doi:10.1007/s00122-012-1986-4.

John J, Williams E (1998). "T -Latinized Designs." *Australian & New Zealand Journal of Statistics*, **40**(1), 111–118. ISSN 1369-1473, 1467-842X. doi:10.1111/1467-842X.00012.

John JA, Williams ER (1995). *Cyclic and Computer Generated Designs*. Second edition. Chapman and Hall, London.

Kempton RA (1982). "The Design and Analysis of Unreplicated Field Trials." *Vortrage fur Pflanzenzuchtung*, **7**, 219–242.

Kiefer J (1974). "General Equivalence Theory for Optimum Designs (Approximate Theory)." *Annals of Statistics*, **2**, 849–879.

Kirkpatrick S, Gelatt CD, Vecchi MP (1983). "Optimisation by Simulated Annealing." *Science*, **220**, 671–680.

Lynch M, Walsh B (1998). *Genetics and Analysis of Quantitative Traits*. Sinaeur Associates, Sunderland, Massachusetts.

Martin RJ (1986). "On the Design of Experiments Under Spatial Correlation." *Biometrika*, **73**, 247–277.

Martin RJ, Eccleston JA (1992). "Recursive Formulae for Constructing Block Designs with Dependent Errors." *Biometrika*, **79**, 426–430.

Martin RJ, Eccleston JA (1997). "Construction of Optimal and Near Optimal Designs for Dependent Observations Using Simulated Annealing." *Research Report 479/97*, Department of Probability and Statistics, University of Sheffield.

Meuwissen THE, Luo Z (1992). "Computing Inbreeding Coefficients in Large Populations." *Genetics, Selection and Evolution*, **24**, 305–315.

Pagnacco G, Jansen GB (2001). "Use of Marker Haplotypes to Refine Covariances Among Relatives for Breeding Value Estimation." *Journal of Animal Breeding and Genetics*, **118**, 69–82.

Patterson HD, Thompson R (1971). "Recovery of Interblock Information When Block Sizes Are Unequal." *Biometrika*, **31**, 545–554.

Randall M, Abramson D (1997). "An Empirical Study of State Encoding in Tabu Search." *Technical report*, School of Computing and Information Technology, Faculty of Science and Technology, Griffith University, Australia.

Robinson GK (1991). "That BLUP is a Good Thing: The Estimation of Random Effects." *Statistical Science*, **6**, 15–51.

Searle SR (1982). *Matrix Algebra Useful for Statistics*. Wiley Inter Science.

Smith AB, Cullis BR (2018). "Design Tableau: An Aid to Specifying the Linear Mixed Model for a Comparative Experiment." *Technical Report 5-18*, University of Wollongong.

Smith AB, Lim P, Cullis BR (2006). "The Design and Analysis of Multi-Phase Plant Breeding Experiments." *The Journal of Agricultural Science*, **144**(5), 393–409. ISSN 0021-8596.

Taillard E (1991). "Robust Taboo Search for the Quadratic Assignment Problem." *Parallel Computing*, **17**, 443–455.

Thompson R (1985). "A Note on Restricted Maximum Likelihood Estimation with an Alternative Outlier Model." *Journal of the Royal Statistical Society Series B*, **47**, 53–55.

Wang J (2002). "An Estimator for Pairwise Relatedness Using Molecular Markers." *Genetics*, **160**, 1203–1215.

Wilkinson GN, Rogers CE (1973a). "Symbolic Description of Factorial Models for Analysis of Variance." *Applied Statistics*, **22**, 392–399.

Wilkinson GN, Rogers CE (1973b). "Symbolic Description of Factorial Models for Analysis of Variance." *Applied Statistics*, **22**, 392–399.

Williams ER, John JA (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 ER, John JA, Whitaker DR (2006). "Construction of Resolvable Spatial Row-Column Designs." *Biometrics*, **62**, 103–108.

Woodruff DL, Zemel E (1993). "Hashing Vectors for Tabu Search." *Annals of Operations Research*, **41**, 123–137.

Affiliation:

David Butler
The University of Wollongong
Northfields Avenue
Wollongong 2522, Australia
E-mail: dbutler@uow.edu.au