Manual:

Phenotypic analysis of plant breeding trials

Joseph O. Ogutu, Torben Schulz-Streeck and Hans-Peter Piepho

University of Hohenheim
Department of Crop Science
Fg. Bioinformatics (340c)

Version 1.0.
October 2011

Table of contents

Summary

This manual presents a brief introduction to statistical analysis of phenotypic data derived from replicated, partially replicated and unreplicated experimental designs most commonly used in plant breeding trials, as well as from a series of trials. Key elements of trial designs, including genotypes, plots, blocks, locations and trials, and their relationships to each other and the basic statistical models for analysing results of experiments based on selected common designs are introduced and described. The manual provides some insights and practical tips on when it is appropriate to represent genotypes, blocks or locations as random or fixed factors in statistical models for plant breeding experiments. Illustrative example analyses for the selected designs, based on empirical data sets, are provided, as are generously annotated program codes for implementing the example analyses in the MIXED procedure of the Statistical Analysis System (SAS) and in the R-Asreml package. The manual also highlights the statistical analysis of genotype × environment interaction and stage-wise analysis of phenotypic data. This version (Version 1.0) of the manual is still a work in progress as it were. We intend to expand the manual to encompass the transition from phenotypic data analysis to genomic selection once the pertinent maize marker data set for the Synbreed Project becomes available. We also intend to further develop the material on analysis of genotype × environment interaction and on modelling the variance heterogeneity associated with such interaction. The planned expansion will pay more particular attention to modelling genotype × environment interactions using a variety of variance-covariance structures, including those that account for variance heterogeneity, especially for a series of experiments conducted at multiple locations. We welcome constructive comments and suggestions on what could still be added to the manual or how it may be improved.

1 Introduction

This manual explains how to do phenotypic analysis of field trial data from plant breeding experiments in a step-by-step fashion and how to feed the information resulting from the analysis into genomic selection. A brief introduction to the statistical analysis of phenotypic data from plant breeding experiments, with a strong focus on mixed models is presented. Our focus on mixed models reflects both their popularity in phenotypic and genomic analyses and the current focus of our research. The manual is thus not intended to be either comprehensive or exhaustive. The presentation emphasizes the key practical considerations and challenges in the analysis of phenotypic data and illustrates these using examples based exclusively on real experiments. These include the salient considerations in selecting particular mixed models, based on several factors including the design of breeding trials. Accordingly, we present the key elements of typical plant breeding experimental designs including trials, replicates, blocks and plots and clarify their relationships. We accomplish this using several examples of designs commonly used in plant breeding experiments and showing how trials, replicates, blocks and plots are often organized to generate specific designs. The four basic designs we examine in some detail are the alpha, lattice, augmented and augmented p-rep designs. For each design we present its characteristic structural layout, formulation of a pertinent mixed model and an example empirical dataset. Moreover, we provide extensive example code for analysing the selected phenotypic data using the MIXED procedure of SAS and R-Asreml. Each code is annotated in sufficient detail for ease of comprehension and immediate application.

A crucial issue when using mixed models relates to deciding which effect should be treated as fixed, random or both. We reflect on some considerations and offer practical suggestions that can aid such decisions, particularly for blocks, locations, genotypes and location-genotype interactions. Lastly, we consider the analysis of a series of experiments.

2 Experimental designs in plant breeding trials

Statistical analysis of phenotypic data in plant breeding is crucially dependent on the design of experiments used to generate the data. As a result, understanding the logic and structure underlying experimental designs commonly used in plant breeding is basic to understanding and effectively using statistical models for phenotypic data analyses. We therefore first take a brief and informal look at some of the designs commonly used in plant breeding trials as a basis for motivating the statistical models used for phenotypic data analysis in § 2.3.

In early generation testing, normally many entries (also known as varieties, cultivars or genotypes) are available for testing. But the number of entries that can be tested is often limited by space and other resources necessary for the proper conduct of trials even though the expected genetic gain may be larger for selections based on screening many genotypes than for those based on more precise assessments of a small subset of genotypes (Kempton and Fox, 1997). A practical compromise often made involves the use of unreplicated trials. Moreover, incomplete blocks are often used when many entries are tested to minimize the error variance within blocks. As well, replicated checks (standard varieties) are often used to enable estimation of the error variance between plots and to adjust for local environmental effects. Commonly used designs which satisfy these constraints include the augmented and augmented p-rep designs. Yet other widely used designs in this regard are the alpha and lattice designs. We therefore first briefly introduce each of these four designs and the statistical models typically used to analyse phenotypic data derived from each design in section § 2.3. But before we delve into the details of these designs it is helpful to introduce,

define and clarify the essential building blocks of plant breeding experiments and how they relate to one another.

2.1 Trials, replicates and blocks and their relationships in plant breeding experiments

Several plots can be grouped together into complete blocks or incomplete blocks. As with plots, several blocks (complete or incomplete) can be grouped into replicates, which, in turn, can be grouped to form trials. A key reason for such grouping is to account for variability between different environments within a location. Another reason is that it is often not possible to sow or harvest all genotypes on the same day. All incomplete blocks within replicates in which cultivars were sown or harvested on the same day may thus be grouped together into one trial to take account of the variation in sowing or harvest dates. Figure 1 presents a schematic layout of a lattice design illustrating the relationships between plots, blocks, replicates and trials (Piepho et al., 2006).

Trial 1											
	Replicate 1						Replicate 2				
	Block 1	Block 2	Block 3	Block 4	Block 5		Block 1	Block 2	Block 3	Block 4	Block 5
Plot 1	5	14	17	4	13		6	C1	2	3	9
Plot 2	C4	8	9	20	18		C5	10	7	C3	19
Plot 3	16	2	C2	10	11		15	13	16	11	12
Plot 4	6	19	15	C3	C5		4	17	20	1	18
Plot 5	3	C1	1	12	7		14	5	C2	8	C4

Trial 2											
	Replicate 1						Replicate 2				
	Block 1	Block 2	Block 3	Block 4	Block 5		Block 1	Block 2	Block 3	Block 4	Block 5
Plot 1	C3	29	C2	26	21		24	C3	C1	32	C4
Plot 2	38	C1	23	32	C5		37	22	26	35	38
Plot 3	27	37	30	24	34		C5	40	27	C2	31
Plot 4	28	25	31	40	33		39	30	23	21	33
Plot 5	35	36	39	C4	32		28	25	34	29	36

Figure 1. A schematic diagram showing sets of five incomplete blocks grouped into two replicates and sets of the two replicates grouped into two trials in a lattice design. Each incomplete block has four entries (i.e., genotypes or cultivars) and one check (C1-C5).

2.2 Types of common experimental designs used as examples in this manual

Experimental designs in plant breeding can be classified into (1) replicated trials and (2) unreplicated trials each of which encompasses a variety of designs. The replicated designs are exemplified by the randomized complete block, alpha and lattice designs. The unreplicated or partially replicated trials, on the other hand, are typified by the augmented design with replicated checks and the augmented p-rep design in which checks are replaced with entries of interest. Table 1 presents the generic design types, specific examples of each and links to basic statistical models applicable to each design.

Table 1: Types of experimental designs used in examples for this manual.

Replicated/ unreplicated	Type of experimental design	Basic model
Replicated	Randomized complete block design (§ 2.4.1)	Table 2
	Resolvable incomplete block designs	
	- Alpha-designs (§ 2.4.2)	Table 3
	- Lattice designs (§ 2.4.3)	Table 3
Unreplicated	Augmented design, checks in blocks (§ 2.5.1)	Table 2
	Augmented p-rep design, replicated entries in blocks (§ 2.5.2)	Table 3

2.3 Basic statistical models for common types of plant breeding trials

Breeders typically conduct a trial (TRL) at multiple locations (LOC). A single trial usually comprises a limited number of entries (GEN). In order to accommodate many entries, several trials are often conducted side-by-side in the same locations. Appropriate statistical models for analysing plant breeding trials thus vary understandably with the trial design. For ease of application, we identify several stylized models and the generic types of trial designs (Table 1) for which they are appropriate. Tables 2 and 3 give an overview of the basic models used for analysis depending on the type of trial design, the number of trials (single or several) and locations (single or several) under consideration. While Table 2 considers trials with one level of blocking, Table 3 looks at trials with two levels of blocking. When analysing several locations, it is crucial to always add a location × genotype interaction effect, because from experience, such interactions are known to be almost always present. Conversely, for a given location, it is usually assumed that blocking factors (TRL, REP and BLK = block) do not interact with treatments (GEN).

Table 2: One level of blocking (BLK = block) per trial (TRL) and location (LOC). A forward slash (/) between two factors means that the factor after the slash is nested within the factor before the slash. So, for example, TRL/BLK means that blocks are nested within trials. A dot (.) between two factors, such as TRL.BLK, means a crossed effect and not an interaction effect that would also require both the TRL and BLK main effects to also be included in the model. Consult Piepho et al. (2003) for further details on how to use these operators.

Type of experiment	Basic model
Single location, single trial	BLK + GEN
Single location, several trials	TRL/BLK + GEN = TRL + TRL.BLK + GEN
Several locations, single trial	LOC/BLK + GEN + LOC.GEN = LOC + LOC.BLK + GEN + LOC.GEN
Several location, several trials	LOC/TRL/BLK + GEN + LOC.GEN = LOC + LOC.TRL + LOC.TRL.BLK + GEN + LOC.GEN

Table 3: Ttwo levels of blocking (REP = replicate and BLK = block within REP) per trial (TRL) and location (LOC). See the caption to Table 2 for explanation of the meanings of the operators used in this table.

Type of experiment	**Base model**
Single location, single trial	REP/BLK + GEN = REP + REP.BLK + GEN
Single location, several trials	TRL/REP/BLK + GEN = TRL + TRL.REP + TRL.REP.BLK + GEN
Several locations, single trial	LOC/REP/BLK + GEN + LOC.GEN = LOC + LOC.REP+ LOC.REP.BLK + GEN + LOC.GEN
Several location, several trials	LOC/TRL/REP/BLK + GEN + LOC.GEN = LOC + LOC.TRL + LOC.TRL.REP + LOC.TRL.REP.BLK + GEN + LOC.GEN

2.4 Replicated designs

2.4.1 Randomized complete block design

This is the most widely used design in agricultural experiments. In this design plots (experimental units) with similar characteristics are arranaged together into groups or blocks. Genotypes are assigned to the plots within blocks such that each genotpe occurs the same number of times, typically only once, within each block. Equivalently, each genotype is replicated the same number of times in each block. A key aim of this design is to minimize the variation among plots within blocks and maximize the variation among blocks. Achieving this goal requires not only grouping similar plots to form blocks but also applying the same techniques to all the plots within a block over the course of the experiement. If required, variation of techniques or experimental conditions likely to alter the experimental outcome is often only made between blocks. A variety of criteria are used to group plots into blocks, including forming a block from a square compact group of adjacent plots. This design can be used to control for a gradient in one direction. Statistical analysis of data from a randomized complete block design is rather simple and hence has been omitted. For a large number of genotypes, complete blocking is well known to be utterly inefficient. Hence most breeders will use some form of incomplete blocking rather than a randomized complete block design.

Block 1	Block 2	Block 3	Block 4
5	4	2	6
4	5	6	1
6	6	4	4
2	3	3	2
1	2	1	3
3	1	5	5

Figure 2. Schematic illustation of a randomised complete block design with four blocks, each of which has six varieties (or genotypes). Each of the six varieties is assigned to one plot.

2.4.2 Alpha-design

An alpha-design is an incomplete block design, in which the blocks can be grouped into complete replicates. Such designs, which encompass small, incomplete blocks within each complete replicate are said to be "resolvable". John and Williams (1995) provide an example of an alpha design based on results from a yield trial involving oats. The example features a trial with 24 genotypes, three complete replicates and six incomplete blocks nested within each replicate. Each block has a size of four, that is, contains four plots.

The layout of their example alpha design is as follows (Fig. 3):

Replicate 1						Replicate 2						Replicate 3					
Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
11	21	23	13	17	6	8	24	12	5	2	19	11	2	17	12	21	3
4	10	14	3	15	12	20	15	11	9	18	7	1	15	18	13	22	5
5	20	16	19	7	24	14	3	21	10	13	6	14	9	4	10	16	20
22	2	18	8	1	9	4	23	17	1	22	16	19	8	6	23	24	7

Figure 3. A sample layout of the alpha design of 24 genotypes reported in John and Williams (1995).

An appropriate model for this design must have an effect for the complete replicates, and another effect for the incomplete blocks, nested within replicates.

Such a model can be specified as

$$y_{ijh} = \mu + \gamma_j + b_{jh} + \tau_i + e_{ijh} \qquad (1)$$

where

y_{ijh} = yield of the *i*-th genotype in the *h*-th block nested within the *j*-th complete replicate

μ = general effect or mean

γ_j = effect of the *j*-th complete replicate

b_{jh} = effect of the *h*-th block nested within the *j*-th complete replicate

τ_i = effect of the *i*-th genotype

e_{ijh} = residual plot error associated with y_{ijh}.

2.4.2.1 Example SAS code 1

This example uses yield data from a trial involving oats based on the alpha design reported in John and Williams (1995). The SAS code below reads these data into SAS and fits model 1 in SAS Proc Mixed. SAS creates a data file called alpha (line 1) with four variables (2) and reads in the following 72 observations (3). The symbol @@ in (2) instructs SAS to continue reading until it reaches the end of a line before moving to the next line. The Proc mixed statement invokes the Mixed procedure (4) and directs it to use the data file called alpha (4). The class statement declares replicates, blocks and genotypes as categorical variables (5). The model statement instructs Proc Mixed to fit model 1 (6). The expression rep*block in line (6) does not specify an interaction between replicates and blocks as such, rather it enables SAS to construct unique identifiers for blocks nested within replicates. Hence, this expression can be substituted with a single variable that uniquely identifies all the blocks used in the trials. The lsmeans statement instructs Proc Mixed to compute the adjusted means for each of the 72

genotypes (7) and to use the SAS output delivery system (ods) to output the adjusted means into a file called lsmeans_gen (8) which can be found in the SAS work library. The run statement tells Proc mixed that all the instructions required to execute the analysis have been provided and instructs it to execute the requested analysis (9). The example can be copied, pasted and run in SAS. It is worth noting that when any of the variables contains characters and not numbers then its name should be followed by the dollar sign (e.g. block2 $, if block2 is variable containing character strings).

```
*1*;   data alpha;
*2*;   input rep   block     gen     y @@;
*3*;   datalines;

 1       1       11    4.1172    1       1        4    4.4461   1       1        5    5.8757
 1       1       22    4.5784    1       2       21    4.6540   1       2       10    4.1736
 1       2       20    4.0141    1       2        2    4.3350   1       3       23    4.2323
 1       3       14    4.7572    1       3       16    4.4906   1       3       18    3.9737
 1       4       13    4.2530    1       4        3    3.3420   1       4       19    4.7269
 1       4        8    4.9989    1       5       17    4.7876   1       5       15    5.0902
 1       5        7    4.1505    1       5        1    5.1202   1       6        6    4.7085
 1       6       12    5.2560    1       6       24    4.9577   1       6        9    3.3986
 2       1        8    3.9926    2       1       20    3.6056   2       1       14    4.5294
 2       1        4    4.3599    2       2       24    3.9039   2       2       15    4.9114
 2       2        3    3.7999    2       2       23    4.3042   2       3       12    5.3127
 2       3       11    5.1163    2       3       21    5.3802   2       3       17    5.0744
 2       4        5    5.1202    2       4        9    4.2955   2       4       10    4.9057
 2       4        1    5.7161    2       5        2    5.1566   2       5       18    5.0988
 2       5       13    5.4840    2       5       22    5.0969   2       6       19    5.3148
 2       6        7    4.6297    2       6        6    5.1751   2       6       16    5.3024
 3       1       11    3.9205    3       1        1    4.6512   3       1       14    4.3887
 3       1       19    4.5552    3       2        2    4.0510   3       2       15    4.6783
 3       2        9    3.1407    3       2        8    3.9821   3       3       17    4.3234
 3       3       18    4.2486    3       3        4    4.3960   3       3        6    4.2474
 3       4       12    4.1746    3       4       13    4.7512   3       4       10    4.0875
 3       4       23    3.8721    3       5       21    4.4130   3       5       22    4.2397
 3       5       16    4.3852    3       5       24    3.5655   3       6        3    2.8873
 3       6        5    4.1972    3       6       20    3.7349   3       6        7    3.6096
;
run;

*4*;   proc mixed data=alpha;
*5*;   class rep block gen;
*6*;   model y=rep rep*block gen;
*7*;   lsmeans gen;
*8*;   ods output lsmeans=lsmeans_gen;
*9*;   run;
```

Using the input statement as indicated in example SAS code 1 is a convenient means for reading small data sets directly into SAS but can be both tedious and time-consuming for relatively large data sets likely to be encountered in practice. For such data sets it is more convenient and efficient to use the SAS import wizard or the SAS import procedure described in more detail in § 11.1 in the Appendix.

2.4.2.2 Example R-Asreml code 1

An equivalent analysis to that done in SAS above can be carried out in R-Asreml by submitting the following call to the Asreml package, within the R software environment:

```
alpha <- data.frame(

rep=factor(c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,
3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3)),

block=factor(c(1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,6,6,6,6,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,6,6,6,6,1,1,1,1,
2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,6,6,6,6)),
```

```
gen=factor(c(11,4,5,22,21,10,20,2,23,14,16,18,13,3,19,8,17,15,7,1,6,12,24,9,8,20,14,4,24,15,3,23,12,11,21,17,5,
9,10,1,2,18,13,22,19,7,6,16,11,1,14,19,2,15,9,8,17,18,4,6,12,13,10,23,21,22,16,24,3,5,20,7))

y=c(4.1172,4.4461,5.8757,4.5784,4.654,4.1736,4.0141,4.335,4.2323,4.7572,4.4906,3.9737,4.253,3.342,4.7269,4
.9989,4.7876,5.0902,4.1505,5.1202,4.7085,5.256,4.9577,3.3986,3.9926,3.6056,4.5294,4.3599,3.9039,4.914,3.79
99,4.3042,5.3127,5.1163,5.3802,5.0744,5.1202,4.2955,4.9057,5.7161,5.1566,5.0988,5.484,5.0969,5.3148,4.629
7,5.1751,5.3024,3.9205,4.6512,4.3887,4.5552,4.051,4.6783,3.1407,3.9821,4.3234,4.2486,4.396,4.2474,4.1746,4
.7512,4.0875,3.8721,4.413,4.2397,4.3852,3.5655,2.8873,4.1972,3.7349,3.6096)
 )
```

```
alpha.asr <- asreml(
        fixed=y ~ gen + rep + rep:block ,
        data = alpha)
```

For routine applications, it is more convenient and quicker to import data sets into R from foreign formats, such as excel, as explained in § 11.2 in the Appendix .

2.4.3 Lattice design

Lattice designs (Fig. 4) are an important class of incomplete block designs for plant breeders. These designs share with the alpha design the common property that randomization and analyses are done in exactly the same way! These designs have the following two key properties. The number of entries must be an exact square. The square root of the number of entries equals (1) the number of blocks and (2) the number of plots in each block. Several incomplete blocks are grouped together to form separate complete replications. Lattice designs are either balanced or partially balanced depending on the number of replications they require. In balanced lattices all entries occur in the same block an equal number of times. In partially balanced lattices not all entries occur together in the same block.

Replicate 1		
Block 1	Block 2	Block 3
C1	3	C2
2	C3	4
6	1	5

Replicate 2		
Block 1	Block 2	Block 3
C1	2	6
3	C3	1
C2	4	5

Replicate 3		
Block 1	Block 2	Block 3
C1	C2	3
C3	2	4
5	1	6

Replicate 4		
Block 1	Block 2	Block 3
C1	3	C2
4	2	C3
1	5	6

Figure 4. Example layout of a lattice design with 4 replicates, 9 entries and a block size of 3. Six entries (1-6) and three checks (C1-C3) were used.

It is noteworthy that replicated trials laid out as lattices or augmented alpha designs are often conducted side by side in the same location and need to be analyzed jointly. The alpha and lattice designs are routinely connected by common checks. Joint analysis requires a further blocking factor up the hierarchy: trial / rep / block / plot (Table 3).

2.5 Unreplicated / Partially replicated designs

2.5.1 Augmented design

An augmented design uses replicated checks (standard varieties) but unreplicated entries. A complete block design is built for the checks and blocks are augmented with entries. Thus, the blocks are incomplete with respect to the entries. In the example below (Fig. 5) C1-C3 are the checks and 1-30 are the entries. Overall, 6 incomplete blocks each with a block size of 8 (plots) are used.

Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
C1	30	C3	23	02	C1
14	C3	18	09	C2	29
26	04	27	06	21	07
C3	15	C1	C3	C1	C3
17	C1	25	C2	C3	01
C2	03	28	20	10	C2
22	C2	05	11	08	12
13	24	C2	C1	16	19

Figure 5. An example of the augmented design.

A suitable model for this design must have effects for incomplete blocks. The model is

$$y_{ij} = \mu + b_j + \tau_i + e_{ij} \quad (2)$$

where

y_{ij} = yield of the *i*-th genotype in the *j*-th block

μ = general effect

b_j = effect of the *j*-th block

τ_i = effect of the *i*-th genotype

e_{ij} = residual plot error associated with y_{ij}

2.5.1.1 Example SAS code 2

```
proc mixed data=augmented  lognote ;
class  block gen ;
model y=  block gen;
lsmeans gen;
ods output lsmeans=lsmeans;
run;
```

2.5.1.2 Example R-Asreml code 2

```
asr_Augmented <- asreml(
        fixed=y ~ gen + block,
        data=Augmented)
```

2.5.2 Augmented p-rep design

The p-rep design (Fig. 6) is similar to the augmented design except that the checks are replaced with partially replicated entries. These designs are particularly useful when trials are repeated across locations. When there are *l* locations, one can replicate 1/*l*-th of the entries at each location (i.e., $p = 1/l$) such that each entry is tested with two replicates in one of the *l* locations.

One option for the p-rep design is to build an alpha-design with the replicated entries and then augment the blocks with the unreplicated entries, resulting in one variant of p-rep design called augmented p-rep design by Williams, Piepho and Whitaker (2011). In the example location shown in Fig. 5 below, entries 1-9 (shown in bold face) are replicated whereas entries 10-39 are not.

Replicate 1				Replicate 2		
Block 1	Block 2	Block 3		Block 1	Block 2	Block 3
1	15	**4**		13	34	**5**
23	**9**	32		37	**9**	27
36	30	28		14	10	31
8	39	**6**		**8**	**7**	**1**
24	**5**	12		**2**	**6**	38
7	21	33		16	11	**3**
29	**2**	25		18	20	17
35	22	**3**		**4**	26	19

Figure 6. An illustrative layout of the augmented p-rep design.

A suitable model for the p-rep design is identical to that for the alpha-design, meaning that the analysis of data from a p-rep design proceeds similarly to that for data from an alpha design. The model is:

$$y_{ijh} = \mu + \gamma_j + b_{jh} + \tau_i + e_{ijh} \qquad (3)$$

where

y_{ijh} = yield of the *i*-th genotype in the *h*-th block nested within the *j*-th complete replicate

μ = general effect

γ_j = effect of the *j*-th complete replicate

b_{jh} = effect of the *h*-th block nested within the *j*-th complete replicate

τ_i = effect of the *i*-th genotype

e_{ijh} = residual plot error associated with y_{ijh}.

3 When should genotype, block or location effect be taken as fixed or random?

Depending on the nature and objective of particular statistical analyses, genotypes, locations or blocks can be treated as either fixed or random factors in a statistical model for phenotypic analysis. Making this decision correctly in practice can sometimes be non-trivial and quite challenging yet the decision arrived at can have important consequences for data analysis and its outcome. Here we take a brief look at some of the key considerations that phenotypic data

analysts should make in deciding whether to treat any of these three factors as either fixed or random in an analysis.

3.1 Genotypes

Genotypes (commonly called entries) can be treated either as fixed or random effects in statistical analysis of plant breeding trial data depending on the goal of analysis. If the goal is to obtain adjusted entry means in a phenotypic analysis, then it suffices to take entries as fixed effects and estimate their adjusted means. If the goal of analysis is to model the variance-covariance matrix of genotypic effects then it is more appropriate to treat the entries as random effects in a phenotypic analysis of plant breeding data. But, if performing genomic selection is the overarching goal of an analysis then entries should be treated as fixed effects in the first stage, and then as random effects in the second stage of a two-stage analysis. In the first stage, the phenotypic analysis stage, the entries are treated as fixed effects and their adjusted means obtained. Once the adjusted entry means are available from the first stage they are treated as the response variable in the second stage (genome-wide analysis stage) in which the entries themselves are treated as random effects. The use of such prediction methods as the widely used Best Linear Unbiased Prediction (BLUPs) to estimate the genomic breeding values of the entries (see e.g., Piepho et al. Submitted) is anchored on the specification of entries as random in the second stage of a two-stage analysis, which will yield BLUPs of genotypic values of entries. Thus, genomic selection can be usefully viewed as a stage-wise approach, in which the first stage corresponds to the phenotypic analysis and the second stage to genomic selection.

In analysis of animal breeding data, which often lack phenotypic observations on individual entries, in contrast, the numerator relationship matrix is routinely used to model genetic relationship among the entries. To obtain estimated breeding values (EBVs) for entries, the entries are specified as random effects. The EBVs are de-regressed and submitted to subsequent steps of step-wise genomic selection. This approach is not recommended for plant breeding data for which direct and replicated phenotypic information on all individual entries is usually available.

3.2 Blocks

The specification of blocks as fixed or random effects in a model determines the type of information that can be extracted from the model and can partly be decided by whether blocks are randomized in a breeding trial or, alternatively, by the number of blocks used in a trial. Thus, if blocks are specified as random then both intra-block and inter-block information can be extracted. But if blocks are used only as fixed effects then only the intra-block information can be acquired. If blocks are randomized then the factor block can be treated as a random effect. Since a large number of blocks are often needed to reliably estimate block variance, the estimate of block variance may not be reliable if only a few blocks are available. Often, however, using blocks as either fixed or random effects yields fairly similar results (Piepho et al., 2003), meaning that data analysts are free to decide which specification to use with their data.

3.3 Locations

As with blocks, the same consideration should be made when deciding whether to specify location as a fixed or random effect in a model. Location should be treated as a random factor, if the locations used in a trial are regarded as a random sample from a target population of environments. This view is in agreement with most plant breeding programs, where the

primary objective is not to breed for one specific location, but for a target population of locations / environments. One advantage of using location as a random factor is that the inter-location information can be gained. Despite this, there are instances when too few locations are available. This often renders the estimation of the location variance unreliable. Also, it may lead to a minor expected increase in information from using location as a random effect, or, worse still, the expected information gain may fail to materialize at all. In some instances, furthermore, using too few locations as random effects may be counterproductive and may actually increase rather than reduce the variance of the difference between entries, contrary to expectation.

Even though location or environment main effect can be treated as random factors (effects) due to random sampling of a subset of locations from a large population of possible locations, location main effect is sometimes treated as a fixed effect, in particular when breeders target a specific population of location or environments. In practice, location main effect is almost always taken as a random effect. Generally when a factor, such as location main effect is random, then conventionally, all other effects that contain that factor, are also taken as random. Hence, for example, when location is taken as a random factor, then all the interactions between location and other factors, such as the location × genotype effect, and all effects nested within location, such as replicates and blocks, also become random effects in the model. If statistical analysis involves at least two-stages, then location is almost never specified as a fixed factor in phenotypic data analysis. It is thus always useful to make a clear distinction between the two different but closely interrelated roles that location can play in models for phenotypic data in plant breeding *viz*: (i) the location main effect and (ii) the genotype-location interaction. The latter should be taken as random always, while for the former there is also the option to formally take it as fixed, as already mentioned above and detailed below (§ 4.1). While there is usually a choice to treat the location main effect as fixed or random, depending on whether or not there is interest in exploiting the between-location information, the genotype-location interaction should always be treated as a random effect, whenever the ultimate goal is to estimate genotype means across locations.

4 Series of experiments

4.1 Basic model

Entries are normally tested in several different locations giving rise to series of experiments. In each location an experimental design, such as one of the designs described above (§ 2.4.1 to 2.5.2), is used. The use of multiple locations enables the analysis to be expanded to cover all the target population of locations. This is achieved by adding a location main effect to the basic model for a single location. Additionally, if it is significant, an interaction term between genotypes and locations should be included. However, it is almost always prudent to include the genotype-location interaction term in the model because it is very rare for this interaction to be insignificant in practice. Otherwise, breeders would not be replicating trials in multiple environments. The main reason for such multiple-location testing is that breeders know, from accumulated experience, that genotype-environment interaction is often substantial.

An example model for an alpha-design pertaining to multiple locations is as follows:

$$y_{ijhk} = \mu + \beta_k + \gamma_{kj} + b_{kjh} + \tau_i + (\tau\beta)_{ik} + e_{ijhk} \qquad (4)$$

where

y_{ijhk} = yield of the *i*-th genotype in the *h*-th block nested within the *j*-th complete replicate in the *k*-th location
μ = general effect
γ_{kj} = effect of the *j*-th complete replicate in the *k*-th location
b_{kjh} = effect of the *h*-th block within the *j*-th complete replicate in the *k*-th location
τ_i = effect of the *i*-th genotype
β_k = effect of the *k*-th location
$(\tau\beta)_{ik}$ = interaction between the *i*-th genotype and the *k*-th location
e_{ijhk} = residual plot error associated with y_{ijk}

It is worth reiterating at this point that if the location factor is treated as random, then conventionally all effects associated with location are also regarded as random. For example, in model 4 above the location main effect, the genotype × location interaction effect, and the block effect that is nested within locations would all be treated as random effects. One can thus make the following distributional assumptions concerning the effects of blocks, locations, their interactions and residual error for model 4:

$$b_{jk} \sim N\left(0, \sigma_b^2\right)$$
$$\beta_k \sim N\left(0, \sigma_\beta^2\right)$$
$$(\tau\beta)_{ik} \sim N\left(0, \sigma_{\tau\beta}^2\right)$$
$$e_{ijk} \sim N\left(0, \sigma_e^2\right)$$

4.1.1 Example SAS code 3

This example SAS code fits data from different locations using an alpha design in each location. The SAS mixed procedure is invoked and instructed to use the dataset called alpha located in the work library or directory (1). The class statement lists replicate, block, genotype and location as classification (i.e. categorical) variables (2). The model statement specifies that the model y = general mean +fixed effect of genotype be fit to the data (3). The random statement lists the intercept, replicate, blocks nested within replicates and genotypes as random effects and location as the subject (4). This is equivalent to fitting location, location*replicate, location*block*replicate and genotype*location as random effects. The adjusted means for each genotype are requested by the lsmeans statement (5) and are output into a file called lsmeans_gen located in the work library (6) after proc mixed executes the submitted commands on reaching the run statement (7).

```
*1*;  proc mixed data=alpha;
*2*;  class rep block gen loc;
*3*;  model y= gen;
*4*;  random int rep block*rep gen/sub=loc;
*5*;  lsmeans gen;
*6*;  ods output lsmeans=lsmeans_gen;
*7*;  run;
```

4.1.2 Example R-Asreml code 3

The preceding analysis can also be performed in R-Asreml using the following code. The operator : between two factors e.g. loc:rep means that the factors location and replicates are

crossed. A crossed effect is not the same as an interaction term. It is only equivalent to an interaction term (e.g. loc*rep) if both loc and rep main effects are also contained in the model.

```
asr <- asreml(
        fixed=y ~ gen,
        random=~loc + loc:rep + loc:rep:block + loc:gen ,
        data = alpha)
```

4.2 Heterogeneity of variance among locations

Oftentimes, differences in precision between locations are quite pronounced in practice, and thus need to be explicitly accounted for. If variance heterogeneity exists, then some gain in efficiency may, be made by accounting for it. This can be done in several ways. Nonetheless, it is sometimes justified to assume homogeneity of variance among locations. For example, model 4 implies homogeneity of variances among locations. While this may seem too strong an assumption, it can be justified by randomization theory (Calinski et al., 2005).

One way to allow for heterogeneity in variances among locations (1 to *k*) using model 4 is to extend it to allow each of the *k* locations to have a different error variance as follows:

$$e_{ijk} \sim N\left(0, \sigma^2_{e_k}\right),$$

4.2.1 Example SAS code 4

This code is identical to that in example 3 above except for the addition of the repeated statement in line 5 that instructs Mixed to fit a separate residual variance for each location through the group=location option.

```
*1*;  proc mixed data=alpha;
*2*;  class rep block gen loc;
*3*;  model y= gen;
*4*;  random int rep block*rep gen/sub=loc;
*5*;  repeated /group=loc;
*6*;  lsmeans gen;
*7*;  ods output lsmeans=lsmeans_gen;
*8*;  run;
```

4.2.2 Example R-Asreml code 4

The heterogeneous location error variance can also be fitted in R-Asreml using the code below. An important new feature in this code is the *rcov* formula used to specify the variance-covariance structure of the residuals (*e*) through the *at()* function for declaring conditional factors. A conditional factor is a factor that is present only at a particular level of another factor. As an illustration, consider a multi-environment trial conducted at two sites using a randomised complete block design in each site. When analyzing these data, one could estimate separate block variance components for each Site by including the random term at(Site):block in the model specification. In this example, Site (= location) is the conditioning factor and separate variance components for block are obtained at each of the two levels of Site (i.e. trial sites or locations). If no levels of the conditioning factor (Site in this example) are specified in the at() function, then a complete set of the conditioning levels is generated. In the present example the at(Site):Block function expands to at(Site,1):Block + at(Site,

2):Block. This is the same as fitting a diagonal variance-covariance model using the function diag(Site):Block. More generally, if the vector (*l*) containing the levels of the conditioning factor (*f*) is specified as a numeric vector then it refers to the levels of *f* in the order returned by levels (*f*). When used in an *rcov* formula as done here, *at()* specifies a variance-covariance model for the residuals (*e*) as a direct sum of *l* variance matrices, one for each level of the conditioning factor. Units refer to the plots on which the yield measurements are made.

```
asr <- asreml(
    fixed=y ~ gen,
    random=~loc+loc:rep+loc:rep:block+loc:gen ,
    rcov = ~ at(loc):units,
    data = alpha)
```

5 Genotype × Environment interaction (to be extended)

Genotype by environment interaction is a fairly common phenomenon in plant breeding trials. Such interactions need to be explicitly accounted for in statistical models for phenotypic analysis of plant breeding data. Several options exist for modelling G × E interaction, such as that in model 4 above, as extended in § 4.2. As an illustrative example, we focus here on the genotype × location part of the following model,

$$\eta_{ij} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} , \qquad (5)$$

where η_{ij} is the conditional expectation of the *i*-th genotype in the *j*-th location. In model 4 we assumed:

$$\beta_j \sim N\left(0, \sigma^2_\beta\right)$$
$$(\tau\beta)_{ij} \sim N\left(0, \sigma^2_{\tau\beta}\right)$$

These assumptions imply a relatively simple variance-covariance structure for η_{ij}, known as "**compound symmetry**", equivalent to assuming a constant correlation between all pairs of genotypes:

$$\text{var}\left(\eta_{ij}\right) = \sigma^2_\beta + \sigma^2_{\tau\beta}$$
$$\text{cov}\left(\eta_{ij}, \eta_{i'j}\right) = \sigma^2_\beta$$

In the terminology of repeated measures experiments, the locations in a series of trials can be viewed as subjects on which repeated measurements are taken and genotypes as the time points at which the measurements are made. Viewed in this way, the compound-symmetric variance-covariance matrix for 4 cultivars may thus be representation by the matrix:

$$\text{var}\begin{pmatrix} \eta_{1j} \\ \eta_{2j} \\ \eta_{3j} \\ \eta_{4j} \end{pmatrix} = \begin{pmatrix} \sigma^2_\beta + \sigma^2_{\tau\beta} & \sigma^2_\beta & \sigma^2_\beta & \sigma^2_\beta \\ \sigma^2_\beta & \sigma^2_\beta + \sigma^2_{\tau\beta} & \sigma^2_\beta & \sigma^2_\beta \\ \sigma^2_\beta & \sigma^2_\beta & \sigma^2_\beta + \sigma^2_{\tau\beta} & \sigma^2_\beta \\ \sigma^2_\beta & \sigma^2_\beta & \sigma^2_\beta & \sigma^2_\beta + \sigma^2_{\tau\beta} \end{pmatrix}$$

The whole variance-covariance matrix has a block diagonal structure. For example, for 4 genotypes and 2 locations we have the following matrix representation:

	Location 1				Location 2				
	Gen 1	Gen 2	Gen 3	Gen 4	Gen 1	Gen 2	Gen 3	Gen 4	
η_{11}	$\sigma_\beta^2+\sigma_{\tau\beta}^2$	σ_β^2	σ_β^2	σ_β^2	0	0	0	0	Location 1
η_{21}	σ_β^2	$\sigma_\beta^2+\sigma_{\tau\beta}^2$	σ_β^2	σ_β^2	0	0	0	0	
η_{31}	σ_β^2	σ_β^2	$\sigma_\beta^2+\sigma_{\tau\beta}^2$	σ_β^2	0	0	0	0	
η_{41}	σ_β^2	σ_β^2	σ_β^2	$\sigma_\beta^2+\sigma_{\tau\beta}^2$	0	0	0	0	
η_{12}	0	0	0	0	$\sigma_\beta^2+\sigma_{\tau\beta}^2$	σ_β^2	σ_β^2	σ_β^2	Location 2
η_{22}	0	0	0	0	σ_β^2	$\sigma_\beta^2+\sigma_{\tau\beta}^2$	σ_β^2	σ_β^2	
η_{32}	0	0	0	0	σ_β^2	σ_β^2	$\sigma_\beta^2+\sigma_{\tau\beta}^2$	σ_β^2	
η_{42}	0	0	0	0	σ_β^2	σ_β^2	σ_β^2	$\sigma_\beta^2+\sigma_{\tau\beta}^2$	

6 Single- vs. Two-stage analysis

Phenotypic analysis can be carried out in either one or two stages each of which has it merits and demerits. For example, a single-stage analysis, though often desirable may sometimes be computationally too demanding. Thus, it is often interesting and useful to proceed in two stages instead. In the first step, adjusted means are computed for each cultivar in each location based on a model such as:

$$y_{ijh} = \mu + \gamma_j + b_{jh} + \tau_i + e_{ijh} \qquad (6)$$

where

y_{ijh} = yield of the *i*-th genotype in the *h*-th block nested within the *j*-th complete replicate

μ = general effect

γ_j = effect of the *j*-th complete replicate

b_{jh} = effect of the *h*-th block nested within the *j*-th complete replicate

τ_i = effect of the *i*-th genotype

e_{ijh} = residual plot error associated with y_{ijh}

These means are then subjected to further analysis by a two-way model. An example two-way model is:

$$\hat{\bar{y}}_{ik} = \mu + \tau_i + \beta_k + (\tau\beta)_{ik} + e_{ik} \qquad (7)$$

where

$\hat{\bar{y}}_{ik}$ = adjusted means of the *i*-th genotype in the *k*-th location

μ = general effect

τ_i = effect of the *i*-th genotype

β_k = effect of the *k*-th location

$(\tau\beta)_{ik}$ = interaction between the *i*-th genotype and the *k*-th location

e_{ik} = adjusted error of the genotype-location mean associated with y_{ik}

In the first step we may compute the variance of a mean ($\hat{\bar{y}}_{ik}$) and use these to model errors e_{ij} in stage two. In stage two, block effects are then implicitly accounted for as random effects, because block means per location are confounded with location main effects.

6.1 Example SAS code 5

Since parts of sample codes 1 to 3 also reappear in this code, we only highlight those parts of the code not covered in the previous examples. The single stage analysis is done in only one step. The lognote option (1) in the single stage analysis prompts proc mixed to write periodic notes to the log window describing the current status of what it is computing. This is useful because computations for the single stage analysis can require extensive computing resources. The parms statement (5) supplies initial values for the four covariance parameters estimated by the mixed model, comprising the location, rep*location, block*rep*location and genotype*location random effects. The estimated covariance parameters are saved in a file called cp_single_stage in the work library (7).

In the first stage of the two-stage analysis, adjusted means for each genotype are computed separately for each location using the by location processing statement (2) and saved in a file known as lsmeans in the work library (7). These lsmeans can then be read (11) into a new file (10) from which the variances of the lsmeans can be extracted (13). The reciprocal of the variance of each lsmean (14) can then be used as its weight in the second stage. The lsmeans are renamed as yield and used as a response variable in the second stage of the analysis (12).

In the second stage, the adjusted mean yield for each genotype over all locations is computed (3), with location and genotype*location as the random effects (4). The parms statement supplies initial values for the location and genotype*location random effect and holds the value of the residual error fixed at 1 using the hold=3 option on the parms statement. Fixing the residual variance at 1 ensures that SAS does not use it to modify the weight assigned to the adjusted mean for each genotype passed on to proc mixed by the weight statement (6).

```
/*---Single-stage Analysis ---------------------*/
*1*; proc mixed data=a lognote;
*2*; class gen loc rep block;
*3*; model y=gen;
*4*; random int rep block*rep gen/sub=loc;
*5*; parms (1)(1)(1)(1)(1);
*6*; lsmeans gen;
*7*; ods output lsmeans=lsmeans_single_stage covparms=cp_single_stage;
*8*; run;

/*---Two-Stage Analysis-------------------------*/

/*---First stage----------*/
*1*;  proc mixed data=a lognote;
*2*;  by loc;
*3*;  class gen rep block;
*4*;  model y=gen;
*5*;  random rep block*rep;
```

```
*6*;  parms (1)(1);
*7*;  lsmeans gen;
*8*;  ods output lsmeans=lsmeans covparms=cp;
*9*;  run;

*10*; data lsmeans;
*11*; set lsmeans;
*12*; y=estimate;
*13*; var_mean=stderr**2;
*14*; w=1/var_mean;
*15*; run;

/*-----Second stage-----------*/
*1*;  proc mixed data=lsmeans lognote;
*2*;  class gen loc;
*3*;  model y=gen;
*4*;  random int gen/sub=loc;
*5*;  parms (1)(1)(1)/hold=3;
*6*;  weight w;
*7*;  repeated ;
*8*;  lsmeans gen;
*9*;  ods output lsmeans=lsmeans2 covparms=cp2;
*10*; run;
```

The weighted method is explained only for SAS because R-Asreml is usually very fast and hence does not benefit much from a stage-wise analysis.

7 Example phenotypic analysis for a lattice design with two replicates (KWS, Synbreed Project dataset)

This example features the case of several lattice experiments (up to 10 × 10 lattices) conducted in six locations (Fig. 7). The dataset was generated for the Synbreed Project. The Synbreed dataset contains 1500 entries tested using a 10 × 10 lattice design with two replicates. All the trials were conducted at four of the six locations. All the 16 trials were conducted at two of the six locations but varying subsets of the trials were conducted at the other four locations. The trials employed 5 to 6 checks (commercial hybrids). Moreover, three different testers were used, but each genotype was tested against only one of the three testers.

Replicate 1											Replicate 2									
B1	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**B8**	**B9**	**B10**		**B11**	**B12**	**B13**	**B14**	**B15**	**B16**	**B17**	**B18**	**B19**	**B20**
7	39	40	27	19	22	44	52	**C2**	**C3**		94	58	19	100	42	68	77	99	12	38
26	10	49	38	29	13	23	34	**C1**	24		73	14	43	61	67	83	6	85	76	53
46	**C4**	8	20	48	45	35	43	**C5**	51		64	89	9	29	25	**C3**	86	10	36	18
36	50	28	9	37	33	55	14	54	32		47	46	35	72	**C1**	26	98	75	59	48
17	18	16	47	6	53	15	25	12	42		84	3	65	33	2	93	13	45	24	62
66	77	69	60	97	71	73	63	82	72		56	69	22	44	15	49	34	37	50	28
88	100	86	78	68	2	95	74	75	85		**C6**	32	54	88	90	39	**C5**	55	81	92
57	58	79	70	76	91	83	84	5	92		**C4**	78	87	20	57	60	23	27	4	74
80	67	59	98	56	64	3	93	94	**C6**		17	**C2**	80	52	97	5	66	63	70	82
99	87	96	90	89	81	62	4	61	65		40	91	96	8	79	71	51	16	95	7

Figure 7. Layout of the lattice design used in each location for each trial/lattice

7.1 Example SAS code 6

Due to the high number of observations, fitting a model to the complete phenotypic data is not feasible in SAS because SAS simply runs out of memory. Hence the data for only a subset of the locations can be analyzed using the proc mixed procedure of SAS.

```
proc mixed data=test lognote;
ods output lsmeans=lsmeans CovParms=covparms FitStatistics=fit;
class gen loc trial rep block;
model y=gen /ddfm=residual notest;
random int trial trial*rep trial*rep*block gen/sub=loc;
parms (1)(1)(1)(1)(1)(1);
lsmeans gen;
run;
```

7.2 Example R-Asreml code 6

The dataset can be analysed with R-Asreml using the following model:

```
########--- Baseline model
asr_a <- asreml(fixed=y ~ gen ,
            random=~loc + loc:trial + loc:trial:rep +
                  loc:trial:rep:block + loc:gen,
            data = a)

######--Model 1: Heterogeneous error variance among the locations
asr_a_model1 <- asreml(fixed=y ~ gen,
                random=~loc + loc:trial + loc:trial:rep +
                      loc:trial:rep:block + loc:gen,
                rcov = ~ at(loc):units,
                data = a)
```

Information-theoretic model selection criteria such as AIC (a smaller AIC values indicates a better model) can be used to choose between the two contending error variance models.

8 Example phenotypic analysis for an augmented design (AgReliant dataset)

The other example dataset contains 177 un-replicated double haploid maize (*Zea mays*) lines each derived from a biparental cross (Fig. 8). The hybrid performance was tested with one common tester. The hybrids were tested using an augmented design. In each of 6 locations, 5 incomplete blocks were used. The dataset was unbalanced so that not every hybrid was tested in each environment. Each incomplete block contained a single column of plots. The incomplete blocks in each environment were connected by two standard varieties (checks) in each incomplete block. The standard varieties were not replicated within the blocks.

Block 1	Block 2	Block 3	Block 4	Block 5
Check 1	Check 1	Check 1	Check 1	Check 1
Check 2	Check 2	Check 2	Check 2	Check 2
DHL 1	DHL 36	DHL 71	DHL 106	DHL 140
.	.	.	.	.
.	.	.	.	.
DHL 35	DHL 70	DHL 105	DHL 140	DHL 177

Figure 8. Layout of the experimental design used in each location. The entries are shown here in alphabetical order for clarity but were randomized in the trials.

A suitable model for this design is:

$$y_{ihk} = \mu + b_{kh} + \tau_i + \beta_k + (\tau\beta)_{ik} + e_{ihk} \quad (11)$$

where

y_{ihk} = yield of the *i*-th genotype in the *h*-th block nested within the *k*-th location
μ = general effect
b_{kh} = effect of the *h*-th block within the *k*-th location
τ_i = effect of the *i*-th genotype
β_k = effect of the *k*-th location
$(\tau\beta)_{ik}$ = interaction between the *i*-th genotype and the *k*-th location
e_{ihk} = residual plot error associated with y_{ihk}

We note that modelling heterogeneous variances between the different environments is difficult due to lack of replicates within locations. Nevertheless, a heterogonous error variance for location appeared better supported by AIC for both datasets.

8.1 Example SAS code 7

Proc mixed invokes the mixed procedure and identifies a dataset called *g* located in the default work library (i.e., work.g) as the dataset to be analysed (1). Location, block and genotype are listed in the class statement as classification variables (2). The model statement fits the mean model: *yield=intercept +fixed genotype effect* (3). To accelerate computations, the denominator degrees of freedom (ddfm) is set to residual via the ddfm=residual option and tests for significance of fixed effects are suppressed through the notest option of the model statement (3). The random statement fits random effects for intercept, block and genotype and specifies location as the subject; which is the same as fitting random effects for location, location*block and location*genotype (4). The adjusted means for the genotypes is requested via the lsmeans statement (5) and estimates of the adjusted means are saved in a file called lsmeans in the work directory. In addition, fit statistics, including the Akaike Information Criterion, Schwarz Bayesian Information Criterion and the log likelihood of the fitted model are requested through the fitstatistics option of the SAS output delivery system and saved in a filed called fit in the work directory (6). The estimated covariance parameters are also saved in a file called *cp* in the work library (6). The parms statement (7) instructs proc mixed to set the initial value for each of the four covariance parameters (viz, the three random effects—location, location*block and location*genotype—and one residual error term) to 1 (6). The numerical values listed in the parms statement must appear in the same order as the covariance parameters in the model.

This model can be extended by the addition of the repeated statement in line 9, which fits a separate residual error variance for each location as dictated by the group=location option.

```
/*--------- Homogeneous location error variance model--------*/

*1*; proc mixed data=g;
```

```
*2*; class loc block gen;
*3*; model y= gen /ddfm=residual notest;
*4*; random int block gen /sub=loc;
*5*; lsmeans gen;
*6*; ods output lsmeans=lsmeans FitStatistics=fit covparms=cp;
*7*; parms(1)(1)(1)(1);
*8*; run;

/*--------- Heterogeneous location error variance model--------*/
    proc mixed data=g;
    class loc block gen;
    model y= gen /ddfm=residual notest;
    random int block gen/sub=loc;
*9*; repeated /group=loc;
    lsmeans gen;
    ods output lsmeans=lsmeans FitStatistics=fit covparms=cp;
    parms(1)(1)(1)(1);
    run;
```

8.2 Example R-Asreml code 7

```
#--------- Homogeneous location error variance model--------

asr <- asreml(
fixed= y ~ gen,
random=~ loc + loc:block + loc: gen,
data = g)

#--------- Heterogeneous location error variance model--------
asr <- asreml(
        fixed= y ~ gen,
        random=~ loc + loc:block + loc: gen,
        rcov = ~ at(loc):units,
        data = g)
```

9 Genomic selection (still to be written)

Section to be completed when the marker data for the KWS Synbreed data becomes available.
Examples:
-Ridge regression BLUP
-Machine learning

10 References

Caliński T. Czajka S, Kaczmarek Z, Krajewski P and Pilarczyk W. 2005. Analyzing multi-environment variety trials using randomization-derived mixed models, *Biometrics* **61**: 448–455.

John JA and Williams ER.1995. Cyclic and Computer Generated Designs. Chapman & Hall, London.

Kempton RA and Fox PN. 1997. Statistical methods for plant variety evaluation. Chapman & Hall, London, UK.

Pedersen, R.G. 1994. Agricultural field experiments: Design and analysis. Marcel Dekker, New York.
Piepho, H-P, Büchse, A and Emrich K. 2003. A hitchhiker's guide to mixed models for randomized experiments. Journal of Agronomy and Crop Science, **189**: 310-322.
Piepho, H-P., A. Büchse, A., and Richter, C. 2004. A Mixed Modelling Approach for Randomized Experiments with Repeated Measures. Journal of Agronomy and Crop Science 1**90**, 230-247.
Piepho, H-P, Büchse, A and Trueberg B. 2006. On the use multiple lattice designs and α -designs in plant breeding trials. Plant Breeding, **125,** 523-528.
Piepho H-P, Schulz-Streeck T, Ogutu JO (Submitted). Analysis of multi-environment trials by stage-wise approaches
SAS Institute. 2011. SAS system for windows, Version 9.2. SAS Institute, Cary, NC. USA.
Butler, D.G., Cullis, B.R., Gilmour, A.R., Gogel, B.J. Mixed models for S language environments: ASReml-R reference manual. Version 3. Draft Copy March 2009 http://www.genstat.co.uk/downloads/asreml /release3/asreml-R.pdf
Williams, E.R., Piepho, H.P. and Whitaker, D. (2011): Augmented p-rep designs. Biometrical Journal **53**, 19-27.

11 Appendix

11.1 Importing data into SAS using the SAS import wizard

Using the input statement as indicated in the example SAS code 1 above is a convenient means for reading small data sets directly into SAS but can be both tedious and time-consuming for relatively large data sets likely to be encountered in practice. For such data sets it is more convenient and efficient to use the SAS import wizard or the SAS import procedure. The SAS import wizard allows you to import a data set saved in other formats not native to SAS, such as excel format. Suppose you want to import an excel data file called maize_yield saved at "D:\Maize\yield\data\maize_yield.xls" into SAS work directory (called library) and save it as a SAS file called yield. Suppose further that the excel sheet in this file that you want to import is called Sheet1. You can do this by following the steps below, where the horizontal arrows are used to indicate what follows or what you should expect to see displayed on the screen after the execution of each step is completed.

From the SAS program editor window click once on "File" in the upper left corner→scroll down to "import data" and click once → What type of data do you wish to import? Select a data source from the list below→Ms excel workbook →Click Next→Connect to Ms excel workbook → browse to the file you want to import and click on the file once to select it (e.g. D:\Maize\yield\data\maize_yield.xls)→Click OK→What table do you want to import? (i.e. Scroll down the list and select the excel sheet you want to import from the list, e.g. Sheet1 in this example. Note that SAS will import only one excel sheet at a time) →Choose the SAS destination (i.e. Choose the SAS directory or library in which to save the imported data→work (The default SAS library called work can be used) →member (i.e. provide a SAS name for the imported file, e.g. yield in this example) →Finish.

Instead of clicking finish you may opt to click next instead for the SAS import wizard to automatically create for you a file containing the SAS import procedure statements. As the next and final step you can browse to the location where you would want SAS to save the generated import statements, provide a name for the program file and click finish. You can open and use the saved SAS program statements by double clicking on the file name from

windows explorer or from within the SAS program editor to import the data from excel without having to use the SAS import wizard. The saved SAS import procedure program for our example will be:

```
PROC IMPORT OUT= WORK.yield
        DATAFILE= ". D:\Maize\yield\data\maize_yield.xls "
        DBMS=EXCEL REPLACE;
    RANGE="Sheet1";
    GETNAMES=YES;
    MIXED=NO;
    SCANTEXT=YES;
    USEDATE=YES;
    SCANTIME=YES;
RUN;
```

It is important to emphasize that the SAS work directory is only a temporary directory and all the data files held in it are lost at the end of each SAS session. To save the imported data set permanently in SAS format you can save it either in the SAS USER library or in any other permanent directory. Using the libname statement of SAS is one convenient way of doing this. For example to permanently save the file work.yield in a directory called yieldata outside SAS, you can use these two statements:

```
libname yielddata "D:\Maize\yield\data\";
Data yielddata.yield; set yield; run;
```

As with the import wizard, the SAS export procedure wizard can be used to export data from SAS to other foreign format destinations, such as excel, dbf, text files, etc.

11.2 Importing data into R

Assume we want to import data into R. We illustrate here how to import data saved in a tab delimited text file format. To import a data set called *alpha.txt* saved in a folder with a location path *D:\Maize\yield\data*, we first specify the location path of the folder and then read the data into R as follows:

```
setwd("D:/Maize/yield/data ")
a<- read.delim (file=" alpha.txt", header=TRUE, sep=" ")
```

Data saved in other foreign formats can be similarly imported into R by looking up and following the appropriate R import commands.

11.3 Datasets

The example datasets for alpha, lattice and augmented are provided in tab delimited text files called alpha.txt, lattice.txt and augmented.txt, respectively. The Synbreed dataset is provided in a tab delimited text file called Synbreed.txt whereas the AgReliant dataset in a text file called AgReliant.txt.