Theoretical and Applied Genetics
https://doi.org/10.1007/s00122-020-03716-8

ORIGINAL ARTICLE

# Linear models for diallel crosses: a review with R functions

**Andrea Onofri**[1] · **Niccolò Terzaroli**[1] · **Luigi Russi**[1]

Received: 15 July 2020 / Accepted: 23 October 2020


## Abstract

***Key message*** **A new R-software procedure for fixed/random Diallel models was developed. We eased the diallel schemes approach by considering them as specific cases with different parameterisations of a general linear model.**

**Abstract** Diallel experiments are based on a set of possible crosses between some homozygous (inbred) lines. For these experiments, six main diallel models are available in literature, to quantify genetic effects, such as general combining ability (GCA), specific combining ability (SCA), reciprocal (maternal) effects and heterosis. Those models tend to be presented as separate entities, to be fitted by using specialised software. In this manuscript, we reinforce the idea that diallel models should be better regarded as specific cases (different parameterisations) of a general linear model and might be fitted with general purpose software facilities, as used for all other types of linear models. We start from the estimation of fixed genetical effects within the R environment and try to bridge the gap between diallel models, linear models and ordinary least squares estimation (OLS). First, we review the main diallel models in literature. Second, we build a set of tools to enable geneticists, plant/animal breeders and students to fit diallel models by using the most widely known R functions for OLS fitting, i.e. the 'lm()' function and related methods. Here, we give three examples to show how diallel models can be built by using the typical process of GLMs and fitted, inspected and processed as all other types of linear models in R. Finally, we give a fourth example to show how our tools can be also used to fit random/mixed effect diallel models in the Bayesian framework.

## Introduction

A diallel experiment is based on a set of possible crosses between some homozygous (inbred) lines and it is usually aimed at quantifying genetic effects, such as:

1. *General combining ability (GCA)*, that is the discrepancy from the average performance of two parental lines in a hybrid combination. Based on Sprague and Tatum (1942), GCA mainly depends on the additive effects of genes, as well as on additive by additive interaction effects;
2. *Specific combining ability (SCA),* that is the effect by which certain hybrid combinations give relatively better/worse performances, compared to the average performances of their parental lines. SCA is regarded as an indication of loci with non-additive effects and includes dominance and epistatic interaction (additive by dominance and dominance by dominance interaction);
3. *Reciprocal (maternal) effect,* that relates to the discrepancy between the performances of a hybrid, e.g. '$A \times B$' and its reciprocal '$B \times A$'. In some instances, (see Cockerham and Weir 1977), reciprocal effects are partitioned into two components: the *reciprocal general combining ability (RGCA)*, that refers, in general, to a parent line in all its combinations and the *reciprocal specific combining ability (RSCA)*, that refers to a specific combination of two parental lines. Reciprocal effects are of great importance for appropriate selection of parents as male or female in hybrid development (Mahgoub 2011);
4. *Heterosis*, that is the change in performance for crosses, with respect to parental lines.

Communicated by Mikko J. Sillanpaa.

**Electronic supplementary material** The online version of this article (https://doi.org/10.1007/s00122-020-03716-8) contains supplementary material, which is available to authorized users.

✉ Niccolò Terzaroli
niccolo.terzaroli@studenti.unipg.it

[1] Dipartimento di Scienze Agrarie, Alimentari e Ambientali, Universita' Degli Studi di Perugia, Borgo XX Giugno, 74, 06121 Perugia, Italy

Published online: 06 November 2020

The assessment of genetic effects is very useful in plant breeding; for example, a high GCA value can predict a flow of several desirable additive genes from parents to offspring (Franco et al. 2001). Moreover, the same authors showed that a high GCA estimate may indicate high heritability and low environmental effects, which may also result in low gene interactions, high selection response and large adaptability. It was also shown that GCA performances of future generations could be predicted by assessing the GCA of a line in an early generation (Lv et al. 2012), saving time and costs. On the contrary, for traits with non-additive gene action, selection should be undertaken in later generations, when genes will be fixed in the homozygous lines (Fasahat et al. 2015). Genetic effects were also determined to describe non-additive gene effects (Singh et al. 1986; Chigeza et al. 2014) and to identify heterotic groups or patterns (Napolitano et al. 2020).

Genetic effects may be of interest either by themselves or combined in the form of ratios. For example, the GCA/SCA ratio determines which type of gene action is involved: a relatively large ratio indicates the prevalence of additive genetic effects, while a relatively low ratio (e.g. $< 1$) the prevalence of dominance and/or epistatic gene effects (Christie and Shattuck 2010).

In order to estimate the above listed genetic effects, a diallel experiment may be planned by using four types of mating designs (Harriman and Nwammadu 2016), including:

1. Crosses, reciprocals and selfed parents (complete diallel)
2. Crosses and selfed parents (half-diallel: no reciprocals)
3. Crosses and reciprocals (no selfed parents)
4. Only crosses (no selfed parents, no reciprocals)

Furthermore, Griffing (1956) distinguished two different situations:

1. If the interest of the investigator lies only in the lines (parents) involved in the experiment and not beyond these crosses, the genetic effects are considered fixed;
2. If the lines (parents) are a sample from a wider population, the genetic effects are considered random and there is no interest in the effect of each single cross/parental line.

Considering the number of genetic effects, the four mating designs and the two possible models (fixed and random), we can easily understand the reasons why a plethora of estimation methods is available in literature, which may, at first, look overwhelming. For example, Hayman (1954) presented one method for mating design 1, while Griffing (1956) presented 8 different methods (4 mating designs, each with random and fixed effects). In Gardner and Eberhart (1966) two additional methods were introduced, later extended to include multiple environments (Eberhart and Gardner 1966).

What can be dreadful for a beginner is that all these estimation methods are not presented on a common ground. Furthermore, the choice of the appropriate method could be rather difficult, as not all of them can be interchangeably used in all situations. In fact, it has been shown that the estimates of SCA effects in Griffing's methods 1 and 2 may be biased, due to the inclusion of parental lines (Yao et al. 2013). At the same time, Griffing's method 3 was deemed to be the best for estimating SCAs and maternal/reciprocal effects. Nonetheless, Griffing's method 4 is also considered reliable, it requires half of the crossings to be made, and hence, much less effort, especially when the number of parents is high (Acquaah 2012).

In this manuscript, following Möhring et al. (2011), we would like to reinforce the idea that all methods of analysis for diallel data should be seen as specific cases (different parameterisations) of a general linear model; consequently, all diallel models should be built and fitted by using a common platform.

Relating to random genetic effects, Möhring et al. (2011) put all models within the frame of linear mixed models (LMM) and restricted maximum likelihood (REML) estimation. They also proposed a software implementation with SAS and ASREML (Gilmoure et al. 2015). Other mixed model solutions have been proposed by Xiang and Li (2001), Wu and Matheson (2001), Xu and Zhu (1999). Focusing on the R statistical environment (R Core Team 2019), diallel mixed models can be efficiently fitted with the package 'asreml-R' (Butler et al. 2018) and with the free package 'sommer' (Covarrubias-Pazaran 2016). Several other possibilities exist, as listed in the introduction of the manuscript by Covarrubias-Pazaran (2016), while the most widespread packages for mixed models in R, i.e. 'nlme' (Pinheiro et al., 2020) and 'lme4' (Bates et al. 2015) do not appear to have been extensively used for fitting diallel models.

With concern to the fixed genetic effects, the traditional estimation methods proposed by Hayman (1954) and Griffing (1956) provide unbiased and minimum variance estimators only with balanced data. Furthermore, these estimation methods require specific patterns of calculations, whereas it would be desirable to have a general approach that also works for unbalanced data. The estimation of fixed genetic effects should be undertaken within the frame of general linear models, preferably by ordinary least squares (OLS), as already suggested by Gardner and Eberhart (1966). Implementations can be found for the SAS language (see Zhang and Kang 1997; Zhang et al. 2005; Makumbi et al. 2018) and in stand-alone software (Tong et al. 2012). To the best of our knowledge, the availability of tools for diallel analysis in the R statistical environment is rather limited. The already mentioned 'asreml-R' and 'sommer' packages are

tailored to the needs of random effect estimation and only the latter is free to use. Other resources include the package 'DiallelAnalysisR' (Yaseen and Eskridge 2020) and the set of R functions presented in Singh et al. (2015). Both tools are self-contained and are not rooted in the typical frame of ordinary least squares (OLS) estimation in R.

Apart from REML (for mixed effects models) and OLS (for fixed effects models), a third option has recently made its way among geneticists, that is the Bayesian framework. Bayesian methods assume that, before making an experiments, model parameters are characterised by a prior probability distribution, that summarises our previous knowledge about the phoenomenon. After the experiment, the previous knowledge is updated by considering the observed data and it is expressed by way of a posterior distribution, representing our final knowledge about the phenomenon (Kery 2010). The Bayesian framework is very flexible and it can accommodate both random and fixed effects models, with several advantages in terms of interval estimation (Li and Loken 2002). Relating to diallel models, the use of a Bayesian frame for diallel models was advocated by Lenarcic et al. (2012), who created the R package 'BayesDiallel'. An example of application in plant breeding was given by Turner et al. (2018).

It is somewhat surprising to see that, at present, fitting diallel models seems to require specific software and/or packages. In our view, all analyses should be performed by using standard, general purpose linear model fitting software. With focus on the R environment, it would not appear that, at present, fitting diallel models by using the most widespread functions, such as 'lm()', 'lme()' or 'lmer()' seems to be relatively rare. With reference to Bayesian methods, to the best of our knowledge there are no examples of using the very powerful and flexible BUGS environment (Spigelhalter et al. 2003).

This manuscript will initially focus on fixed genetical effects, relating the crosses between specific parents. The reason for this choice is three-fold: first, random/mixed effects diallel models can already be fitted in the general mixed model framework, by using the already mentioned 'asreml-R' package, or by using other general packages, such as 'MCMCglmm' (Hadfield 2010). Second, fixed effect models are useful to test for the significance of genetical effects, which is a common requirement among plant breeders. Last, but not least, while we recognise the flexibility of random effect models, we argue that the estimation of variance components may be unreliable when the number of parents is small. A brief survey of literature shows that mating designs with three (Amin 2015), four (Quimio and Zapata 1990) and five (Singh and Jain 1971; Dhaliwal and Gill 1973) parents are used. Furthermore, standard errors for variance components are seldomly reported; but, to our experience, they may be very high, also with mating designs with eight parents. Focusing on fixed genetic effects, the objective of the present work is to bridge the gap between linear models and diallel models in R. To do so, instead of building a brand new piece of software, we decided to follow another route, that is to build a set of tools which would enable geneticists, plant/animal breeders and students to fit diallel models by the most widely known R functions for OLS fitting. Finally, we will show that this set of tools can be also used to fit fixed/random effect diallel models in the Bayesian framework, by using a very widely known Markov Chain Monte Carlo (MCMC) sampler.

## Methods

### Model definitions

The results of diallel experiments might be described by using the usual two-way ANOVA model, where we consider the factorial combination of $n$ parentals taken either as 'father' or as 'mother':

$$y_{ijk} = \mu + \gamma_k + \alpha_i + \beta_j + \alpha\beta_{ij} + \varepsilon_{ijk} \quad (1)$$

where $y_{ijk}$ is the yield (or any other trait of interest) for the combination between the parents $i$ and $j$ in the block $k$, $\mu$ is the intercept, $\gamma_k$ is the effect of the $k$th block, $\alpha_i$ is the 'paternal' effect of the $i$th 'father', $\beta_j$ is the 'maternal' effect of the $j$th mother and $\alpha\beta_{ij}$ is the interaction effect, describing the non-additive effect of a specific combination between the $i$th father and $j$th mother. The residual error term $\varepsilon_{ijk}$ is assumed to be gaussian and i.i.d, with standard deviation equal to $\sigma$.

Except for the case of a full-diallel design, all the other diallel designs and so the one above is unbalanced; in all cases, the 'father' and 'mother' effects are regarded as two completely different series of treatments, neglecting the idea that they are, indeed, the same genotypes in different combinations. Therefore, there was the need to build more appropriate diallel models. For historical reasons, we will start from the model proposed by Hayman (1954), by slightly changing the notation, to make it more consistent, throughout the manuscript:

$$y_{ijk} = \mu + \gamma_k + g_i + g_j + ts_{ij} + rg_i^a + rg_j^b + rs_{ij} + \varepsilon_{ijk} \quad (2)$$

where $\mu$ is expected value (the overall mean, in the balanced case), $g_i$ and $g_j$ are the GCAs of the $i$th and $j$th parents, $ts_{ij}$ is the total SCA (tSCA) for the combination between the $i$th and $j$th parent, $rg_i^a$ and $rg_j^b$ are the RGCAs for the $i$th and $j$th parents, under the constraint that $rg_i^a = -rg_j^b$ for one specific parent $i$ and $rs_{ij}$ is the RSCA for a specific $ij$ combination, i.e. the discrepancy between the effect of the $i$ and $j$ parents, when they are used as 'father' or 'mother',

respectively. Obviously, reciprocal effects can only be estimated when the experimental design includes the reciprocal crosses.

The four models devised by Griffing (1956) need not be listed, as they can be seen as particular cases of the more general Eq. 2, where the reciprocal effect (REC) is not parted into RGCA and RSCA $\left(r_{ij} = rg_i^a + rg_j^b + rs_{ij}\right)$. Of course, when the reciprocals have not been included in the mating design, the term $r_{ij}$ should be removed from the equation.

According to Hayman (1954), the tSCA effect can be partitioned in three additive components, leading to the following system of equations:

$$y_{ijk} = \begin{cases} \mu + \gamma_k + g_i + g_j + m + d_i + d_j + s_{ij} + rg_i^a + rg_j^b + rs_{ij} + \varepsilon_{ijk} & \text{for } i \neq j \\ \mu + \gamma_k + 2g_i - (n-1)m - (n-2)d_i + \varepsilon_{ijk} & \text{for } i = j \end{cases} \quad (3)$$

where $n$ is the number of parentals, $m$ relates to the difference between the average yield of selfed parents and the average yield of crosses (mean dominance deviation; MDD), the $d$ parameters relate to the differences between the yield of each selfed parent ($Y_{ij}$, with $i=j$) and the average yield of all selfed parents (dominance deviation for the $i$th parent; DD) and $s_{ij}$ is the residual SCA effect for the combination $ij$.

It should be noted that both Eqs. 2 and 3 consider the genetical effects as differences with respect to the intercept $\mu$, that is the mean of all observations. Due to unbalance (the number of crosses is never equal to the number of selfed parents), such an approach requires the introduction of some coefficients (i.e. $n-1$ and $n-2$ in Eq. 3), which do not have an obvious meaning. Additional models were proposed, which do not consider the overall mean as the intercept, but allow for different means for crosses and selfed parents (Gardner and Eberhart 1966). One such model is usually known as GE2, and it may be formulated as:

$$y_{ijk} = \mu_v + \gamma_k + 0.5\left(v_i + v_j\right) + \bar{h} + h_i + h_j + s_{ij} + \varepsilon_{ijk} \quad (4)$$

where $\mu_v$ is the intercept, corresponding to the overall mean for all selfed parents (not the overall mean, as in previous models). The parameters $v$ ($v_i$ and $v_j$) represent the differences between the expected value for the selfed parents $i$ and $j$ and the mean for all selfed parents ($\mu_v$). According to the authors, this would be the variety effect (VE); as a consequence, the expected value for the $i$th selfed parent is $\mu_v + v_i$, while the expected value for the cross $ij$, in the absence of any dominance/heterosis effects, would be $\mu_v + 0.5(v_i + v_j)$, that is the mean value of its parents. There is a close relationship between $g_i$ and $g_j$ in Eqs. 2 and 3 and $v_i$ and $v_j$ in Eq. 4, that is: $v_i = 2g_i + (n-2)d_i$; therefore, the sum of squares for the GCA and VE effects are the same, although the estimates are different.

Since a cross not necessarily responds according to the mean value of its parents, the parameter $\bar{h}$ represents the average heterosis (H.BAR) contributed by the whole set of genotypes used in crosses. In the balanced case, $\bar{h}$ represents the difference between the overall mean for selfed parents and the overall mean for crosses, under the constraint that $\bar{h} = 0$ for all selfed parents. Besides, the parameters $h_i$ represent the average heterosis contributed by the $i$th parent in its crosses (Hi), while $s_{ij}$ is the SCA for the cross between the $i$th and $j$th parents, that is totally equivalent to the corresponding parameter in Eq. 3.

It is clear that both Eqs. 3 and 4 account for the heterosis effect, although they do it in a different way: in Eq. 3 the specific effect of heterosis is assessed with reference to the overall mean, while in Eq. 4 it is assessed by comparing the mean of a cross with the means of its parents. Indeed, the sum of squares for the 'MDD' and 'Hi' effects are perfectly the same, although the parameters are different.

Gardner and Eberhart proposed another model (GE3), which we have slightly modified to maintain a consistent notation in the frame of GLMs:

$$y_{ijk} = \begin{cases} \mu_v + \gamma_k + \bar{h} + \mathrm{gc}_i + \mathrm{gc}_j + s_{ij} & \text{for } i \neq j \\ \mu_v + \gamma_k + \mathrm{sp}_i & \text{for } i = j \end{cases} \quad (5)$$

Equation 5 is an array composed of two separate elements for crosses and selfed parents. For the crosses (equation above), the parameters $gc_i$ and $gc_j$ represent the GCA for the $i$ and $j$ parents in all their crosses (GCAC); it should be noted that GCA ≠ GCAC, as this latter effect is estimated without considering the selfed parents. The parameters $s_{ij}$ are the same as in the previous models (SCA effect), while $sp_i$ represents the effects of selfed parents (SP): they are numerically equivalent to the corresponding effects in Eq. 4, but the sum of squares are different and such a discrepancy has been put forward and discussed by Murray et al. 2003, to whom we refer for further detail. Therefore, we use different names for these two effects (SP and Hi).

## Model implementation

Our view is that all diallel models (Eqs. 2 to 5 and other similar equations) are linear models with different parameterisations. Therefore, it would be useful to be able to fit all those models in R, by using the standard, general purpose facilities for linear model fitting. Focusing on fixed effects, every linear model can be written (in matrix notation) as:

$$y = X\beta + \varepsilon \tag{6}$$

where $y$ is the vector of the observed response, $X$ is the design matrix, $\beta$ is the vector of parameters and $\varepsilon$ is the vector of residuals, assumed as gaussian distributed, with mean equal to 0 and variance equal to $\sigma^2$. The estimation of β is accomplished by minimising the sum of squared residuals (RSS = $\varepsilon^{\mathrm{T}}\varepsilon$), which is possible by the following equation:

$$\beta = (X^T X)^{-1} X^T y \tag{7}$$

The OLS solution is also the maximum likelihood solution, provided that errors are independent and identically distributed. We see that Eq. 7 requires the availability of the design matrix $X$.

In R, the typical linear model fitting function based on OLS is 'lm()', which uses the 'model.matrix()' function to build design matrices, according to the user-defined (or default) parameterisation. The main implementation problem is that certain effects, such as the GCA, require the definition of unconventional design matrices, using algorithms that are not available in R. The packages 'asreml-R' and 'sommer' add a few functionalities, such as the ability of overlaying design matrices (function 'and()' in 'asreml' and 'overlay()' in 'sommer'), which is useful to code GCA effects. However, none of the two packages plays well with the 'lm()' function in R and at present, to the best of our knowledge, there is no simple way to fit diallel models with fixed effects by using the 'lm()' function in R.

Several authors suggested how to build design matrices for half-diallel designs (Wu and Matheson 2000, 2001; Tong et al. 2012). We extended these suggestions to build a handful of new R functions, aimed at producing the correct design matrices for all the above mentioned effects. The syntax is simple; for example, the GCA effect can be specified by using the function '*GCA(Par1, Par2)*', where 'Par1' and 'Par2' are two variables coding for parentals. For all other effects, only the name of the function changes, according to the naming in previous paragraphs (GCA, tSCA, RGCA, RSCA, REC, DD, MDD, H.BAR, Hi, VEi, SP and GCAC).

By using these R functions, we can fit all diallel models inside the 'lm()' and 'lme()' functions. For example, Eq. 3 can be fitted by using the usual code for linear models:

```
lm(yield ~ GCA(Par1, Par2) + tSCA(Par1, Par2), data = df)
```

where 'df' is a 'dataframe' hosting the response and explanatory variables. Similarly, we can introduce the effect of reciprocals by using the following code:

```
lm(yield ~ GCA(Par1, Par2) + tSCA(Par1, Par2)
  + REC(Par1, Par2), data = df)
```

This latter definition corresponds to Griffing's model 1 (Eq. 2, replacing $rg_i^a + rg_j^b + rs_{ij}$ with $r_{ij}$); however, if we were willing to partition the tSCA effect, we could code the following model:

```
lm(yield ~ GCA(Par1, Par2) + MDD(Par1, Par2) + DD(Par1, Par2)
+SCA(Par1, Par2) + REC(Par1, Par2), data = df)
```

which does not correspond to any of the Hayman's, Griffing's or Gardner–Eberhart's models, nonetheless it has relevant potentialities. If we replace '*REC(Par1, Par2)*' with '*RGCA(Par1, Par2)* + *RSCA(Par1, Par2)*' we get Hayman's model 2 (Eq. 3); in case of no reciprocals, we can remove the REC effects altogether, while in case of no reciprocals and no selfed parents, we can build a model with the GCA effect only.

Another possible model is:

```
lm(yield ~ H.BAR(Par1, Par2) + VE.i(Par1, Par2)
  + H.i(Par1, Par2) + SCA(Par1, Par2), data = df)
```

that corresponds to Gardner–Eberhart model 2 (Eq. 4) and could be enhanced by including the effects '*REC(Par1, Par2)*' or '*RGCA(Par1, Par2)* + *RSCA(Par1, Par2)*', when reciprocals are available. Lastly, the GE3 model (Eq. 5) is:

```
lm(yield ~ H.BAR(Par1, Par2) + SP(Par1, Par2)
  + GCAC(Par1, Par2) + SCA(Par1, Par2), data = df)
```

that can, as well, be enhanced by adding reciprocal effects, if necessary.

In summary, we propose that diallel models are flexibly built by using the typical process of model fitting that has become in fashion with GLMs, considering: (i) the information we have at hand (whether we have crosses, selfs and/or reciprocals) and (ii) the effects we want to estimate. In this process, we have only one model frame and different parameterisations, as anticipated above.

However, we do not intend to neglect the importance of referring to some relevant model parameterisations, by using the names of the authors. For this reason, we also built a wrapper function named 'lm.diallel()', which can be used in the very same fashion as 'lm()'. The syntax is:

```
lm.diallel(formula, Block, Env, data, fct)
```

where 'formula' uses the regular R syntax to specify the response variable and the two variables for parentals (e.g., Yield ~ Par1 + Par2). The two arguments 'Block' and 'Env' are used to specify optional variables, coding for blocks and environments, respectively. The argument 'data' is a 'dataframe' where to look for explanatory variables. Finally, 'fct' is a string variable coding for the selected model. In this regard, we considered the main six diallel models in literature: Hayman's model 1 (Eq. 2), Hayman's model 2 (Eq. 3), Griffing's model 1 (Eq. 2, with $r_{ij} = rg_i^a + rg_j^b + rs_{ij}$), Griffing's model 2 (Eq. 2, without

reciprocal effects), Gardner–Eberhart model 2 (Eq. 4) and Gardner–Eberhart model 3 (Eq. 5). For these six models, the 'fct' string should take, respectively, the following values: "HAYMAN1", "GRIFFING1", "GRIFFING2", "HAYMAN2", "GE2", "GE3". The strings "GE2r" and "GE3r" can be used to specify the 'enhanced' GE2 and GE3 models, including the effect of reciprocals (REC).

As an example, the GE3 model can be fitted either by using 'lm()' (as shown above) or by using the following syntax:

lm.diallel(yield ~ Par1 + Par2, data = df, fct = "GE3")

For better clarity, we report a table of the correspondences between the equations, the syntax for the 'lm()' function and the value for the 'fct' argument in the 'lm.diallel()' function (Table 1).

One big advantage of fitting diallel models in R with the 'lm()' function or with the 'lm.diallel()' function is that we can exploit the whole infrastructure for 'lm' objects. In particular, we can make profit, e.g. of the usual 'plot', 'summary' and 'anova' methods. With reference to these latter two methods, we should not forget that, by default, the residual term is used as an estimate of pure error. Although the genetic effects in our approach are fixed, using the residual error term is not necessarily a good way to go. For example, the residual error term may not be available, if we work with means and not with raw field data. Furthermore, if we have collected data about the reciprocals, but we do not want to fit reciprocal effects, the residual term is not a good estimate of pure error.

Therefore, we have coded the 'anova.diallel' and 'summary.diallel' methods that, optionally, allow the user to enter a value for the residual error variance and degrees of freedom. This user-defined value is used in inferences and tests of hypotheses, in place of the residual term.

## Linear/nonlinear functions of model parameters

One problem with diallel models is that the effects may not be orthogonal to each other, which may cause some inconsistencies in estimation (Murray et al. 2003). Therefore, it has been proposed that some meaningful quantities (the forementioned authors cite 'heterosis effects', 'variety effects' and GCA) are directly derived from variety and cross means. In the frame of linear models, it is possible to derive those meaningful quantities by linear and nonlinear functions of model parameters (Bretz et al. 2011), while standard errors can be derived by using the propagation of errors and the delta method (Weisberg 2005). In simple terms, when we combine the values of model parameters to derive some meaningful quantity, e.g. an index of heterosis, we can also combine the standard errors to derive a standard error for that heterosis index. If the function of model parameters is linear, the derived standard error is exact; otherwise, it is only approximate and takes the name of delta standard error. With specific reference to ratios, the Fieller's method can also be used for inferences (Piepho and Emrich 2005).

For example, the parameters of Eq. 4 can be easily used to derive mid-parent heterosis (MPH) and best parent heterosis (BPH) (Li et al. 2018). The two equations are:

$$\mathrm{MPH}(\%) = \frac{\overline{h} + h_i + h_j + s_{ij}}{\mu_v + 0.5 \times \left(v_i + v_j\right)} \tag{8}$$

and:

$$\mathrm{BPH}(\%) = \frac{\overline{h} - 0.5 \times \left(v_i + v_j\right) + h_i + h_j + s_{ij}}{\mu_v + \max\left(v_i, v_j\right)} \tag{9}$$

Other useful measures can be derived in the very same fashion, as linear or nonlinear functions of model parameters.

**Table 1** Correspondence between the Eqs. 2 to 5, the value for the 'fct' string in the 'lm.diallel()' function and the syntax for the 'lm()' function

| Equation | Model name in 'lm.diallel()' | Model notation in 'lm()' |
|---|---|---|
| Equation 2 | HAYMAN1 | Y ~ GCA(Par1, Par2) + tSCA(Par1, Par2) + RGCA(Par1, Par2) + RSCA(Par1, Par2) |
| Equation 2[a] | GRIFFING1 | Y ~ GCA(Par1, Par2) + tSCA(Par1, Par2) + REC(Par1, Par2) |
| Equation 2[b] | GRIFFING2 | Y ~ GCA(Par1, Par2) + tSCA(Par1, Par2) |
| Equation 3 | HAYMAN2 | Y ~ GCA(Par1, Par2) + MDD(Par1, Par2) + DD(Par1, Par2) + SCA(Par1, Par2) + RGCA(Par1, Par2) + RSCA(Par1, Par2) |
| Equation 4 | GE2 | Y ~ H.BAR(Par1, Par2) + VE.i(Par1, Par2) + H.i(Par1, Par2) + SCA(Par1, Par2) |
| Equation 5 | GE3 | Y ~ H.BAR(Par1, Par2) + SP(Par1, Par2) + GCAC(Par1, Par2) + SCA(Par1, Par2) |

[a]Terms are redefined as: $r_{ij} = rg_i^a + rg_j^b + rs_{ij}$

[b]The term $r_{ij}$ is not included

## Extension to random and mixed models

Although we have so far focused on fixed effects, we do not neglect the importance and flexibility of building models with random genetic effects. A multilevel mixed model with $r$ random terms can be written in matrix notation as:

$$y = X\beta + Z_1 b_1 + Z_2 b_2 + \ldots + Z_r b_r + \varepsilon \quad (10)$$

where $Z_1$, $Z_2$ and $Zr$ are the design matrices for random effects and $b_1$, $b_2$, $b_r$ are the vectors of random effects, which are assumed to be independent from each other, independent from the residual error term and gaussian distributed, with means equal to 0 and variances respectively equal to $\sigma^2_{b1}$, $\sigma^2_{b2}$ and $\sigma^2_{br}$ (variance components). In contrast to Eqs. 6, 10 has no immediate closed form solution and therefore, parameter estimates need to be obtained by using some sort of mixed model "solver", such as those available in the EMMREML (Akdemir and Godfrey 2015) and SAMM (Akdemir 2018) packages, just to mention a few. These solvers require the design matrices for fixed and random terms ($X$, $Z_1$, $Z_2$, …, $Z_r$), which can be built by using the forementioned tools, together with the 'model.matrix()' function. We also built the 'model.matrixDiallel()' functions that is simpler to use and shares the same syntax as the 'lm.diallel()' function.

For one of our examples, we used the very popular, general purpose MCMC sampler JAGS with its companion R package 'rjags' (Plummer 2019), which appear to be underutilised in the context of plant breeding, although they are very powerful and flexible. JAGS is rooted in the Bayesian framework and requires prior information about all the estimands. For fixed effects, we used Gaussian priors with means equal to 0 and precisions equal to 0.0001 (the precision is the inverse of the variance and it is used in place of this latter in the definition of mixed models in JAGS). For variance components, we used uniform priors from 0 to 100. The MCMC sampler was used with five chains, 10,000 iterations and a burn-in of 1000 iterations.

## R package

All the above R functions and JAGS model definitions are freely available in a gitHub repository (https://github.com/OnofriAndreaPG/lmDiallel) and they have been used to produce an R package (lmDiallel), which can be freely downloaded from gitHub (the code download and installation are shown below). They were written by the standard R language, and they depend on the basic packages 'base' and 'stat', as well as on the package 'plyr' (Wickham 2011). The engine for the JAGS sampler also needs to be installed, together with the 'rjags' package.

## Results

The functionalities and approach of our package can be better described by using a few examples, relating to different experimental situations.

***Example 1*** As an example of mating design 1, we used the dataset shown in Hayman (1954), concerning the flowering times in *Nicotiana rustica*, in a diallel cross with eight inbred varieties. The design was a randomized complete block design with two replicates and for this analysis, we will regard the block as fixed effect. The dataset is available in the 'lmDiallel' package as 'hayman54'.

Relating to the code in Box 1, the first three lines install (if necessary) and load the 'lmDiallel' package from gitHub (the 'devtool' package is necessary to perform the installation and it must have been already installed in the system). The forth line loads the dataset and the subsequent lines fit the Eq. 2 (model HAYMAN2) and show the ANOVA table. The results are entirely the same as those reported in the original paper (see also the supplemental material in Möhring et al. 2010). Being in the typical R platform for linear models has the great advantage that we can use most of the available methods for linear models. Apart from the 'anova()' method, we can also use the 'summary()' method to retrieve the genetic parameters, as commonly done for linear models in R. In Box 2, we show an excerpt of the output.

From Box 2, we see that some estimates are missing. Indeed, we should not forget that parameter estimation is performed under some restrictions, e.g. $\sum_i j_i = 0$ and $\sum_i k_i = 0$, therefore, $j_8 = -\sum_{i=1}^{7} j_i$ and $k_8 = -\sum_{i=1}^{7} k_i$. Linear functions of model parameters can be built by using the standard facilities in R, such as the 'glht()' function in the 'multcomp' package (Bretz et al. 2011). An example is given in Box 3.

A frequently neglected aspect is that diallel models, as all linear models, should be carefully inspected in relation to the basic assumptions of normality and homoscedasticity of residuals. A graphical inspection can be obtained with the usual 'plot.lm()' method, the result is shown in Fig. 1. We see some slight signs of heteroscedasticity, which we will not address here, although we would like to mention that such a problem might be solved by using stabilising transformations or appropriately modelling the variance–covariance matrix for model residuals (Pinheiro and Bates 2000). The selection of the most appropriate method is still under debate, and both solutions have advantages and drawbacks. We refer the readers to the available literature for multi-environment experiments (e.g. Annicchiarico 2002), and we only emphasize that the

**Box 1** Sample code to fit Eq. 2 to the data in Hayman (1954)

```
# library(devtools)
# install_github("OnofriAndreaPG/lmDiallel")
library(lmDiallel)
data(hayman54)

contrasts(hayman54$Block) <- c("contr.sum")
dMod <- lm(Ftime ~ Block + GCA(Par1, Par2) +
             tSCA(Par1, Par2) + RGCA(Par1, Par2) +
             RSCA(Par1, Par2), data = hayman54)
anova(dMod)
## Analysis of Variance Table
##
## Response: Ftime
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## Block              1    142     142  0.3416   0.56100
## GCA(Par1, Par2)    7 277717   39674 95.1805 < 2.2e-16 ***
## tSCA(Par1, Par2) 28 102238    3651  8.7599 6.656e-13 ***
## RGCA(Par1, Par2)  7   6739     963  2.3097   0.03671 *
## RSCA(Par1, Par2) 21  12373     589  1.4135   0.14668
## Residuals        63  26260     417
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**Box 2** Genetical parameters for the data in Hayman (1954)

```
out <- summary(dMod)
out$coefficients[1:13,]
##                           Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)             162.898437   1.804567  90.2700843 2.381071e-68
## Block1                   -1.054688   1.804567  -0.5844545 5.610017e-01
## GCA(Par1, Par2)g_1       46.195312   3.376036  13.6832990 1.558468e-20
## GCA(Par1, Par2)g_2      -24.585938   3.376036  -7.2824864 6.421946e-10
## GCA(Par1, Par2)g_3       49.632812   3.376036  14.7015049 4.900927e-22
## GCA(Par1, Par2)g_4       18.351563   3.376036   5.4358311 9.415231e-07
## GCA(Par1, Par2)g_5      -20.929687   3.376036  -6.1994855 4.846464e-08
## GCA(Par1, Par2)g_6        2.445313   3.376036   0.7243147 4.715539e-01
## GCA(Par1, Par2)g_7      -44.710937   3.376036 -13.2436192 7.232268e-20
## tSCA(Par1, Par2)ts_1:1   33.710937  12.631971   2.6686998 9.670965e-03
## tSCA(Par1, Par2)ts_1:2  -31.507812   9.022836  -3.4920076 8.819590e-04
## tSCA(Par1, Par2)ts_1:3   12.273438   9.022836   1.3602638 1.785940e-01
## tSCA(Par1, Par2)ts_1:4   27.054688   9.022836   2.9984682 3.880843e-03
```

inspection of model residuals appears to be as fundamental for diallel models as it is for all other linear models.

An alternative, though less flexible, way of fitting the same model is by using the wrapper function 'lm.diallel()', which is shown in Box 4. In this case, we do not have to specify the effects, we only have to indicate what model we want to fit (see also Table 1).

***Example 2*** Very often, diallel experiments are repeated across environments (different locations and/or different years). The method outlined in the previous example can be used to produce separate analyses for each environment. However, if the environment is regarded as a fixed factor, we could be interested in testing the significance of the genetical effects and their interactions with the environment.

In order to show how this can be performed in R, we used the dataset shown in Zhang et al. (2005), reporting the results of a diallel experiment with five parents, in two blocks and two environments. Reciprocal crosses are not considered, and they were deleted from the dataset. The dataset ('Zhang05' in Box 5) contains the 'Par1', 'Par2', 'Env', 'Block' and 'Yield' variables, coding for

**Box 3** Use of the *glht()* function in the *multcomp* package to build linear functions of model parameters

```
library(multcomp)
# Getting missing parameters
k <- matrix(c(0,0,-1,-1,-1,-1,-1,-1, -1, rep(0, 56)), 1, 65, byrow=T)
j_8 <-  summary(glht(dMod, linfct=k))
j_8
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = Ftime ~ Block + GCA(Par1, Par2) + tSCA(Par1, Par2) +
##     RGCA(Par1, Par2) + RSCA(Par1, Par2), data = hayman54)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0  -26.398      3.376  -7.819 7.39e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

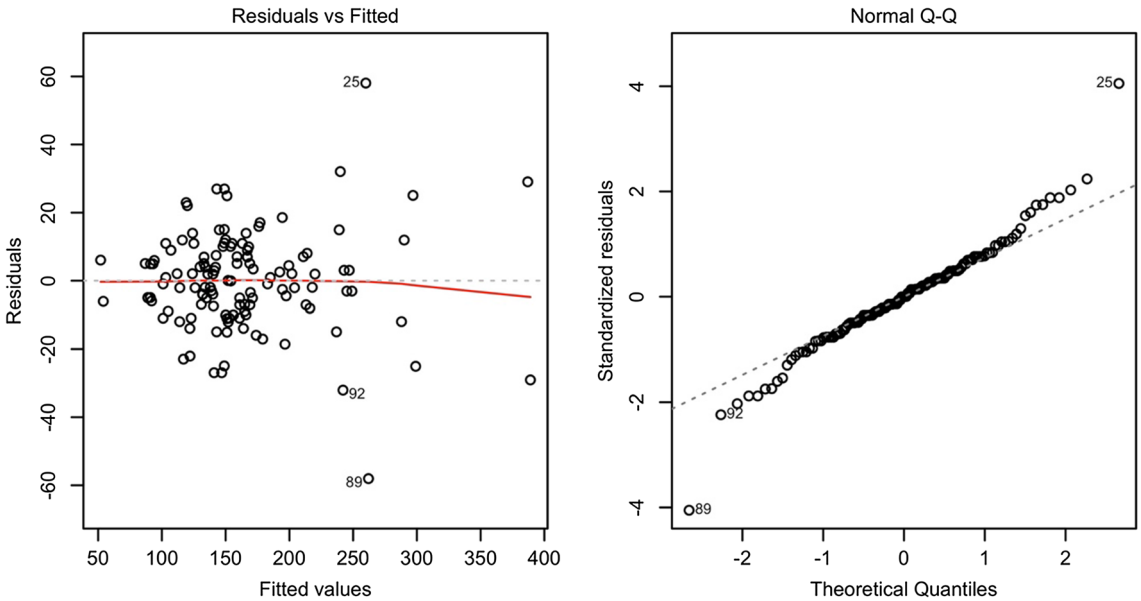


**Fig. 1** Graphical inspection of residuals for a diallel model: plot of residuals against expected values (left) and QQ-plot of standardised residuals (right)

the parents, environments, blocks and yield, respectively. Box 5 shows the code to fit Eq. 4.

The ANOVA table shows that there are no significant interactions between genetical parameters and the environment. Therefore, we can remove those interactions and refit the model to get the average value of genetical parameters.

Functions of model parameters can also be used to retrieve the MPH (%) and BPH (%) (see Eqs. 8 and 9), together with standard errors. In this respect, we can use the 'deltaMethod()' function in the 'car' package (Weisberg 2005). Box 6 gives a simple example for the '1 × 2' cross, which can be easily extended to other crosses by an appropriate script.

Once again, we see that all analyses can be performed by the typical facilities in R, with a little experience in linear modelling. We also like to emphasise that in Box 7 we have removed the environment effects altogether, although in other instances, we might be interested in removing the interaction with the environment only for a subset of genetical effects.

**Box 4** Use of the *lm.diallel()* wrapper to fit the same model as in Box 1

```
dMod2 <- lm.diallel(Ftime ~ Par1 + Par2, Block = Block,
                    data = hayman54,
                    fct = "HAYMAN1")
anova(dMod2)
## Analysis of Variance Table
##
## Response: Ftime
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Block      1    142     142  0.3416   0.56100
## GCA        7 277717   39674 95.1805 < 2.2e-16 ***
## tSCA      28 102238    3651  8.7599 6.656e-13 ***
## RGCA       7   6739     963  2.3097   0.03671 *
## RSCA      21  12373     589  1.4135   0.14668
## Residuals 63  26260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**Box 5** Example of how diallel models can be fit by using the 'lm()' function in R.

```
contrasts(zhang05$Block) <- c("contr.sum")
contrasts(zhang05$Env) <- c("contr.sum")
dMod <- lm(Yield ~ Env/Block + H.BAR(Par1, Par2) + VEi(Par1, Par2) +
                   Hi(Par1, Par2) + SCA(Par1, Par2) +
                   H.BAR(Par1, Par2):Env + VEi(Par1, Par2):Env +
                   Hi(Par1, Par2):Env + SCA(Par1, Par2):Env,
           data = zhang05)
anova(dMod)
```

```
## Analysis of Variance Table
##
## Response: Yield
##                         Df Sum Sq Mean Sq  F value    Pr(>F)
## Env                      1   0.00   0.001   0.0014 0.9703173
## H.BAR(Par1, Par2)        1  39.91  39.905  84.3749 6.065e-10 ***
## VEi(Par1, Par2)          4 630.14 157.535 333.0885 < 2.2e-16 ***
## Hi(Par1, Par2)           4 178.93  44.732  94.5812 7.642e-16 ***
## SCA(Par1, Par2)          5 132.55  26.511  56.0536 1.020e-13 ***
## Env:Block                2   8.99   4.494   9.5013 0.0007088 ***
## Env:H.BAR(Par1, Par2)    1   0.26   0.261   0.5526 0.4634667
## Env:VEi(Par1, Par2)      4   2.08   0.521   1.1019 0.3751175
## Env:Hi(Par1, Par2)       4   0.38   0.094   0.1983 0.9371763
## Env:SCA(Par1, Par2)      5   2.89   0.579   1.2235 0.3242599
## Residuals               28  13.24   0.473
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Alternative definition (with slightly different parameterisation)
dMod2 <- lm.diallel(Yield ~ Par1 + Par2, Env = Env, Block = Block,
                    data = zhang05, fct = "GE2")
anova(dMod2)
summary(dMod2)
```

**Box 6** Using the delta method to retrieve MPH and BPH with standard errors.

```
library(car)
## Loading required package: carData
parNames <- c("muc", "Env2", "h.bar", "v_1", "v_2", "v_3", "v_4",
              "h_1", "h_2", "h_3", "h_4", "s_12", "s_13", "s_14", "s_23",
              "s_24", "Env1:Block1", "Env1:Block1")
MPH12 <- "(h.bar + v_1 + v_2 + s_12)/(muc + 0.5 * (v_1 + v_2))"
BPH12 <- "(h.bar - 0.5*(v_1 + v_2) + h_1 + h_2 + s_12)/(muc + max)"
coefs <- as.list(coef(dMod2))
names(coefs) <- parNames
deltaMethod(dMod2, g = MPH12, parameterNames = parNames, func = "BPH12")
##        Estimate        SE     2.5 %  97.5 %
## BPH12 -0.495408  0.038422 -0.570715 -0.4201
deltaMethod(dMod2, g = BPH12, constants = c(max = max(coefs$v_1, coefs$v_2)),
            parameterNames = parNames, func = "BPH12")
##       Estimate       SE    2.5 % 97.5 %
## BPH12 0.095243 0.031253 0.033988 0.1565
```

**Box 7** Removing the environment effect from the model in Box 5.

```
dMod2 <- lm(Yield ~ Env/Block + H.BAR(Par1, Par2) + VEi(Par1, Par2) +
                  Hi(Par1, Par2) + SCA(Par1, Par2), data = zhang05)
summary(dMod2)$coefficients
##                          Estimate Std. Error      t value     Pr(>|t|)
## (Intercept)          17.416666667  0.1730079 100.66975560 1.040748e-51
## Env1                  0.090000000  0.1537800   0.58500000 0.563060
## H.BAR(Par1, Par2)    -1.730000000  0.1835026  -9.42765792 6.324974e-12
## VEi(Par1, Par2)v_1   -6.395000000  0.2996585 -21.34095724 3.653634e-24
## VEi(Par1, Par2)v_2   -0.670000000  0.2996585  -2.23587824 3.073054e-02
## VEi(Par1, Par2)v_3    0.530000000  0.2996585   1.76867980 8.420963e-02
## VEi(Par1, Par2)v_4    3.130000000  0.2996585  10.44522223 3.007776e-13
## Hi(Par1, Par2)h_1    -1.239166667  0.2288680  -5.41432938 2.748110e-06
## Hi(Par1, Par2)h_2    -0.885000000  0.2288680  -3.86685797 3.768571e-04
## Hi(Par1, Par2)h_3     4.131666667  0.2288680  18.05261941 2.093015e-21
## Hi(Par1, Par2)h_4    -2.535000000  0.2288680 -11.07625418 4.863312e-14
## SCA(Par1, Par2)s_1:2  1.916666667  0.2369009   8.09058506 4.183115e-10
## SCA(Par1, Par2)s_1:3 -1.100000000  0.2369009  -4.64329229 3.359848e-05
## SCA(Par1, Par2)s_1:4 -1.233333333  0.2369009  -5.20611560 5.438735e-06
## SCA(Par1, Par2)s_2:3  0.383333333  0.2369009   1.61811701 1.131240e-01
## SCA(Par1, Par2)s_2:4 -3.050000000  0.2369009 -12.87458318 3.580667e-16
## Env1:Block1          -0.230000000  0.1223351  -1.88008207 6.704465e-02
## Env2:Block1          -0.496666667  0.1223351  -4.05988737 2.094228e-04
```

***Example 3*** This dataset is from a diallel experiment with six maize varieties and no reciprocals (Gardner and Eberhart 1966) and consists of means across blocks, a possibility when we perform the analyses in two-steps: firstly, running separate analyses for all locations and secondly, fitting a model to the entry means. In order to perform the correct inferences, we need an estimate of an appropriate error term, that is usually obtained in the first step; in this example, we used the residual variance (MSE = 7.10 with 60 degrees of freedom), as reported in the paper.

The dataset is available as 'lonnquist61' in the 'lmDiallel' package. The code shown in Box 8 is used to fit a model with GCA effects retrieve the value of estimated parameters. Please, note that the residual variance is passed as an argument to the 'summary()' function, to obtain reliable estimates of standard errors. The estimated parameters and the partitioning of sum of squares are entirely the same as those reported in the paper.

***Example 4*** (**extension to mixed models**) For this final example, we used a multi-environment half-diallel dataset with

**Box 8** Sample code to fit Eq. 4 to the data in Gardner and Eberhart (1966), either with the 'lm()' function or with the 'lm.diallel()' wrapper.

```
data("lonnquist61")
dMod <- lm(Yield ~ H.BAR(Par1, Par2) + VEi(Par1, Par2) +
             Hi(Par1, Par2) + SCA(Par1, Par2),
                  data = lonnquist61)
summary.diallel(dMod, MSE = 7.10, dfr = 60)
##                       Estimate       SE     t value     Pr(>|t|)
## (Intercept)             92.450 1.087811 84.98716970 3.078543e-64
## H.BAR(Par1, Par2)        5.190 1.287116  4.03227173 1.583166e-04
## VEi(Par1, Par2)v_B       4.150 2.432420  1.70611989 9.315754e-02
## VEi(Par1, Par2)v_G      -4.550 2.432420 -1.87056518 6.628496e-02
## VEi(Par1, Par2)v_H      -0.750 2.432420 -0.30833492 7.588955e-01
## VEi(Par1, Par2)v_K      -1.150 2.432420 -0.47278021 6.380853e-01
## VEi(Par1, Par2)v_K2      3.750 2.432420  1.54167460 1.284107e-01
## Hi(Par1, Par2)h_B       -1.175 1.719981 -0.68314723 4.971433e-01
## Hi(Par1, Par2)h_G        0.225 1.719981  0.13081543 8.963589e-01
## Hi(Par1, Par2)h_H        0.350 1.719981  0.20349066 8.394401e-01
## Hi(Par1, Par2)h_K       -2.425 1.719981 -1.40989961 1.637322e-01
## Hi(Par1, Par2)h_K2       4.500 1.719981  2.61630855 1.122922e-02
## SCA(Par1, Par2)s_B:G     4.810 2.063977  2.33045261 2.316255e-02
## SCA(Par1, Par2)s_B:H    -1.415 2.063977 -0.68556974 4.956249e-01
## SCA(Par1, Par2)s_B:K    -0.140 2.063977 -0.06783022 9.461463e-01
## SCA(Par1, Par2)s_B:K2   -2.215 2.063977 -1.07317101 2.874917e-01
## SCA(Par1, Par2)s_G:H    -2.865 2.063977 -1.38809704 1.702404e-01
## SCA(Par1, Par2)s_G:K    -0.990 2.063977 -0.47965657 6.332155e-01
## SCA(Par1, Par2)s_G:K2    1.335 2.063977  0.64680961 5.202220e-01
## SCA(Par1, Par2)s_H:K    -0.515 2.063977 -0.24951832 8.038121e-01
## SCA(Par1, Par2)s_H:K2    1.410 2.063977  0.68314723 4.971433e-01
anova.diallel(dMod, MSE = 7.10, dfr = 60)
## Analysis of Variance Table
##
## Response: Yield
##                    Df  Sum Sq Mean Sq F value    Pr(>F)
## H.BAR(Par1, Par2)   1 115.440 115.440 16.2592 0.0001583 ***
## VEi(Par1, Par2)     5 234.230  46.846  6.5980 5.923e-05 ***
## Hi(Par1, Par2)      5  59.720  11.944  1.6823 0.1527246
## SCA(Par1, Par2)     9  63.781   7.087  0.9981 0.4515416
## Residuals          60           7.100
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# dMod2 <- lm.diallel(Yield ~ Par1 + Par2,
#                     data = lonnquist61,
#                     fct = "GE2")
# summary(dMod2, MSE = 7.10, dfr = 60)
# anova(dMod2, MSE = 7.10, dfr = 60)
```

six parentals, in five blocks and ten environments; the dataset is factitious and was obtained by Monte Carlo simulation, starting from the data shown in Example 2. It is available in the 'lmDiallel' package in the 'diallelMET' data object. We want to fit the Gardner–Eberhart model 2 and as we have a relatively high number of blocks and environments, we are willing to consider these two effects as random. Extending Eq. 10, the model is:

$$\begin{cases} y = X\beta + Z_1 b_1 + Z_2 b_2 + Z_3 b_3 + Z_4 b_4 + Z_5 b_5 + Z_6 b_6 + \varepsilon \\ \varepsilon \sim N(0, \sigma) \end{cases} \tag{11}$$

where $X$ is the design matrix for fixed effects, $\beta$ is the vector of fixed effects; $\varepsilon$ is the vector of model residuals, that is assumed as normally distributed with mean equal to 0 and standard deviation equal to $\sigma$; the number of columns in $X$ is equal to the number of elements in $\beta$, that is 21 (with

five parentals: one column/element for the interaction effect, one for the H.BAR effect, five for the VE.i effect, five for the H.i effect and nine for the SCA effect). The six $Z_i$ matrices and $b_i$ vectors are, respectively, the design matrices for the random effects of 'environments' ($Z_1b_1$, with 9 columns/elements), 'blocks within environments' ($Z_2b_2$, with 40 columns/elements), 'H.BAR by environments' ($Z_3b_3$, with 9 columns/elements), 'VE.i by environments' ($Z_4b_4$, with 45 columns/elements), 'H.i by environments' ($Z_5b_5$, with 45 columns/elements) and 'SCA by environments' ($Z_6b_6$, with 81 columns/elements). All random effects are assumed as gaussian, with means equal to 0 and standard deviations equal to $\sigma_1$, $\sigma_2$, $\sigma_3$, $\sigma_4$, $\sigma_5$ and $\sigma_6$, respectively. As specified in Eq. 11, the residual term $\varepsilon$ is also assumed as gaussian, with mean equal to 0 and standard deviation $\sigma$. As the consequence, there are 28 estimands.

In order to fit the above model with JAGS, we have to specify the definition (in JAGS code) in a text file, as shown in the Supporting Information S1. This definition should contain the likelihood expression for the model to be fitted (Eq. 11), the priors for all estimands ($\beta$ and $b_1$ to $b_6$) and the hyperpriors for the standard deviation parameters ($\sigma$ and $\sigma_1$ to $\sigma_6$). The definition of a JAGS model may require some experience; to ease the task, we provide several template files as a list in the 'bugs_mods' dataset, which can be loaded in R and saved to a text file within the working directory. In Box 9, we show how to load the dataset and JAGS model definition, saving this latter to the 'modelDef.txt' file, to be used in subsequent steps.

Now, we need to create the design matrices, which is feasible by using the 'model.matrix()' function and dismantling the resulting object by way of the 'assign' attribute. We need to know that such an attribute labels the columns belonging to each effect in the model, according to their positioning, from left to right (see Box 10). We also need starting values for the estimands; for fixed effects, we can use the results of a fixed model fit, while for random effects we can set the starting values to 0.1. Care should be taken to ensure that the naming of parameters in the JAGS call correspond to their naming within the model definition file. In the end, we can start the MCMC sampler to obtain samples from the posterior distribution for all the estimated parameters. For the random effects, we request samples for the variance components, although the model was parameterised in terms of the standard deviations.

The model may take some time to fit: in the end, for all estimated parameters we obtain information about the posterior distribution, i.e. the median as measure of central tendency and the credible interval, which is the Bayesian analogue to the confidence interval, although the interpretation is rather different (Onofri 2015). The whole code to perform the analyses is shown in Box 10, where for brevity reasons, we omitted the report about random effects and we only show variance components. Random effects can be simply obtained by appropriately changing the 'init' and 'params' variables in model definition.

## Discussion

Diallel experiments are frequently used to obtain information about the genetical parameters of parental lines. A survey of literature shows that, starting from the mid of the previous century, several models and methods have been proposed, which may be overwhelming for a novice. The most confusing aspect is that all those models and methods are not presented on a common platform, which enhances the perceived difference from one model to another. In this regard, we have tried to reinforce the idea that all diallel models are special cases (different parameterisations) of the same general linear model (see also Möhring et al. 2011).

In order to promote the above view, we have argued that it should be possible to fit diallel models by using the typical frame of OLS estimation, with no need for additional fitting tools. So far, software development has followed the route to building specific tools for diallel analyses; we thought that, instead of building brand new functions for diallel models, it would be relevant to give geneticists and plant/animal breeders the tools to fit diallel models within the general platform of linear models in R.

Linear fixed effect models in R are mainly fitted by using the 'lm()' function and related methods, although at the moment, diallel models cannot be fit by using such function. The problem is that there are no tools to automatically build the design matrices, as implied by the available diallel models. We overcame this gap by building the 'model.

**Box 9** Loading the dataset and model definition.

```
data("diallelMET")
contrasts(diallelMET$Block) <- c("contr.sum")
contrasts(diallelMET$Env) <- c("contr.sum")

data("bugs_mods")
modelDef <- bugs_mods$mod_GE2.ranEnvB
writeLines(modelDef, "modelDef.txt")
```

**Box 10** R code to fit a GE2 model to a multi-environment diallel experiment, with random environments and blocks.

```
# Define design matrices
modMat <- model.matrix(~ H.BAR(Par1, Par2) + VEi(Par1, Par2) +
                  Hi(Par1, Par2) + SCA(Par1, Par2) + Env + Env:Block +
                  H.BAR(Par1, Par2):Env + VEi(Par1, Par2):Env +
                  Hi(Par1, Par2):Env + SCA(Par1, Par2):Env,
                  data = diallelMET)
X <- modMat[,attr(modMat, "assign") <= 4]
Z.1 <- modMat[,attr(modMat, "assign") == 5]
Z.2 <- modMat[,attr(modMat, "assign") == 6]
Z.3 <- modMat[,attr(modMat, "assign") == 7]
Z.4 <- modMat[,attr(modMat, "assign") == 8]
Z.5 <- modMat[,attr(modMat, "assign") == 9]
Z.6 <- modMat[,attr(modMat, "assign") == 10]


# Get starting values for fixed effects
start <- lm(Yield ~ H.BAR(Par1, Par2) + VEi(Par1, Par2) +
                  Hi(Par1, Par2) + SCA(Par1, Par2), data = diallelMET)
betaInit  <- coef(start)

# Create input lists for WinBugs
dataMod <- list(Y = diallelMET$Yield, X = X, Z.1 = Z.1, Z.2 = Z.2,
               Z.3 = Z.3, Z.4 = Z.4,
               Z.5 = Z.5, Z.6 = Z.6)
init <- list(beta = betaInit, sigma = 0.1, sigma.1 = 0.1,
                sigma.2 = 0.1, sigma.3 = 0.1,
                sigma.4 = 0.1, sigma.5 = 0.1,
                sigma.6 = 0.1)

# Start sampler
library(rjags)

mcmc <- jags.model("modelDef.txt",
                  data = dataMod, inits = init,
                  n.chains = 4, n.adapt = 1000)

# Specify the parameters to be sampled from posterior
params <- c("beta", "sigma2", "sigma2.1", "sigma2.2", "sigma2.3",
               "sigma2.4", "sigma2.5", "sigma2.6")
res <- coda.samples(mcmc, params, n.iter = 10000)
out <- summary(window(res, start = 1000))

out$quantiles[,c(1,3,5)]

#                   2.5%           50%         97.5%
# beta[1]   8.152474e+01  81.817269419   82.11561327
# beta[2]   4.920418e+00   5.271833255    5.61847214
# beta[3]   3.974022e+00   4.640349134    5.29568641
# beta[4]  -4.882005e+00  -4.226423884   -3.56131288
# beta[5]  -1.427660e+00  -0.769736742   -0.11815954
# beta[6]  -1.857068e+00  -1.197285852   -0.52314801
# beta[7]   2.667793e+00   3.327177676    3.98646592
# beta[8]  -1.880750e+00  -1.416860166   -0.94428239
# beta[9]  -5.371951e-01  -0.072576178    0.38789486
# beta[10] -1.424521e-01   0.325031377    0.78958610
# beta[11] -3.163768e+00  -2.689559000   -2.22382319
# beta[12]  4.938210e+00   5.403555401    5.86891316
# beta[13] -3.566999e-01   0.197959991    0.75011656
# beta[14] -6.720124e-01  -0.109105633    0.45394006
# beta[15] -2.554556e-01   0.301762536    0.85051718
# beta[16] -1.307294e+00  -0.756305097   -0.20330708
# beta[17] -8.100658e-01  -0.247694484    0.30777565
# beta[18] -5.128433e-01   0.044033401    0.59654472
# beta[19] -2.810446e-01   0.277157395    0.83879972
# beta[20] -1.281412e+00  -0.732947514   -0.17603621
# beta[21] -2.428165e-02   0.532350365    1.08788353
# sigma2    6.166747e+00   6.749335265    7.39922728
# sigma2.1  2.173421e+02 518.532763429 1726.68626146
# sigma2.2  3.887342e+01  59.259940889   95.92964811
# sigma2.3  2.762043e-05   0.017600454    0.28116820
# sigma2.4  3.819699e-04   0.072688487    0.35430244
# sigma2.5  1.488243e-05   0.006312528    0.06564335
# sigma2.6  3.968218e-05   0.003472637    0.04124375
```

matrixDiallel()' function, which is tailored to build the correct model matrix for all the main equations in literature. Possible variations and enhancements (e.g. see Murray et al. 2003; Yao et al. 2013) can be accommodated within the same function.

Once the model matrix is defined, model fitting can be performed by usual tools, such as the 'lm()' function or the 'lme.fit()'. For less experienced users, we built the 'lm.diallel()' function, that is a wrapper to the 'lm.fit()' function with a higher degree of usability, in relation to diallel models. The advantage of both approaches is that they can exploit several powerful methods for linear models, such as 'summary()', anova() or 'glht()' in the 'multcomp' package. In particular, the inspection of model residuals can be made in the very same fashion as for all other linear models, an aspect very frequently neglected in specialised diallel analyses softwares.

Increasing the usability of existing packages that have gained a wide popularity may be an advantageous programming strategy, compared to the usual strategy of building brand new platforms. From the point of view of the developer, it is efficient, as it requires a minor programming effort. From the point of view of the users (professionals, technicians and students), it is handy to be put in the conditions of making statistical analyses, without the need of learning new softwares and/or languages and/or syntaxes. In general, due to its open-source nature, the R environment is often overwhelming for users, that are confused by the extremely wide availability of alternative methods to perform the same task. In this regard, a programming strategy aimed at supporting some existing reference platforms might help build a more comfortable environment for statistical analyses.

One further aspect to be considered is that the dichotomy between random/mixed and fixed diallel models might be regarded as rather outdated; indeed, the availability of REML estimation has opened a third possibility, where we have a specific interest in the lines involved in the experiment, but, nonetheless, we model them as random. It is possible to obtain best linear unbiased predictors (BLUPs) that have been shown to be more accurate than best linear unbiased estimators (BLUEs) of genetical effects (Piepho et al. 2008). It is necessary to point out that good variance component estimates require a relatively high number of parents, which is not often the case with diallel experiments. Therefore, the use of OLS continues to be preferable for experiments conducted with a small number of parent lines, which motivated our initial focus on fixed effects model.

Nonetheless, our tools can also be useful to fit random effects and mixed effects diallel models. In particular, we have shown that it is possible to make use of the design matrices produced as the output of the 'model.matrix()' or 'model.matrixDiallel()' functions within the general purpose MCMC sampler JAGS; our fourth example relates to an experiment where the environment is random, while genetic effects are fixed, although a further extension to models with random genetic effects is immediate. Turner et al. (2018) showed that the Bayesian framework may be advantageous for fitting diallel models, while JAGS represents one of the most flexible and widespread general purpose MCMC samplers around. Both the framework and the sampler still deserve a wider appraisal in plant breeding. Obviously, such aspects as the selection of priors or the check for convergence are fundamental to sound analyses in the Bayesian framework and may require further attention and research by plant breeders. We did not consider these aspects as they appear to be beyond the scope of this manuscript and may require further work.

It is not clear whether and how our tools can be used for REML-based estimation of random diallel models. In principle, the design matrices as used in JAGS can also be used in other REML-based mixed model solvers, which are available within R packages, such as EMMREML (Akdemir and Godfrey 2015) and SAMM (Akdemir 2018), but further research is needed in this respect. For those who favour REML estimation, we provide a few examples on how random diallel models can be fitted in R by using the 'sommer' package (see Supplemental material).

**Acknowledgements** The authors wish to thank Prof. Hans-Peter Piepho (University of Hohenheim, Germany) for critical reading of first version of the manuscript and two anonymous referees for very useful insights and suggestions.

**Author contribution statement** Conceptualization and methodology: A.O., N.T., L.R.; Software preparation, A.O.; Writing - original draft preparation: A.O.; Writing - review and editing: A.O., N.T. L.R. All authors have read and agreed to the published version of the manuscript.

**Funding** Open access funding provided by Università degli Studi di Perugia within the CRUI-CARE Agreement. This research was funded by the Università degli Studi di Perugia (ONOBASERIC18—FFABR).

**Availability of data and material/Code availability** All the above R functions are freely available in a gitHub repository (https://github.com/OnofriAndreaPG/lmDiallel).

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## References


Acquaah G (2012) Principles of plant genetics and breeding. Wiley-Blackwell, Oxford (UK)

Akdemir D (2018) SAMM: some algorithms for mixed models. R package version 1.1.1. Available: https://CRAN.R-project.org/package=SAMM (Date of last access: 09 July 2020)

Akdemir D, Godfrey OU (2015) EMMREML: fitting mixed models with known covariance structures. R package version 3.1. Available: https://CRAN.R-project.org/package=EMMREML (Date of last access: 09 July 2020)

Amin EM (2015) Genetic components and heterotic effect in 3 x 3 diallel crossing experiment on egg production and hatching traits in chickens. J Am Sci 11:140–156

Annicchiarico P (2002) Genotype × environment interactions: challenges and opportunities for plant breeding and cultivar recommendations. FAO plant production and protection paper No. 174. FAO, Rome

Bates D, Mächler M, Bolker B, Walker S (2015) Fitting linear mixed-effects models using lme4. J Stat Softw 67:1–48. https://doi.org/10.18637/jss.v067.i01

Bretz F, Hothorn T, Westfall P (2011) Multiple comparisons using R. CRC Press, Boca Raton

Butler DG, Cullis BR, Gilmour AR, Gogel BJ, Thompson R (2018) ASReml-R reference manual version 4. VSN International Ltd, Hemel Hempstead, HP1 1ES, UK

Chigeza G, Mashingaidze K, Shanahan P (2014) Advanced cycle pedigree breeding in sunflower. II: Combining ability for oil yield and its components. Euphytica 195:183–195. https://doi.org/10.1007/s10681-013-0985-0

Christie BR, Shattuck VI (2010) The diallel cross: design, analysis, and use for plant breeders. Plant breeding reviews. Wiley, New York, pp 9–36

Cockerham CC, Weir BS (1977) Quadratic analyses of reciprocal crosses. Biometrics 33:187. https://doi.org/10.2307/2529312

Covarrubias-Pazaran G (2016) Genome-assisted prediction of quantitative traits using the R package *sommer*. PLoS ONE 11:e0156744. https://doi.org/10.1371/journal.pone.0156744

Dhaliwal HS, Gill AS (1973) Studies of heterosis, combining ability and inheritance of yield and yield components in a diallel cross of Bengal gram (*Cicer arietinum* L.). Theor Appl Genet 43:381–386. https://doi.org/10.1007/BF00278176

Eberhart SA, Gardner CO (1966) A general model for genetic effects. Biometrics 22:864–881. https://doi.org/10.2307/2528079

Fasahat P, Rajabi A, Rad J, Derera J (2015) Principles and utilization of combining ability in plant breeding. Biom Biostat Int J 4:1–22. https://doi.org/10.15406/bbij.2016.04.00085

Franco M, Cassini S, Oliveira V, Vieira C, Tsai SM, Cruz CD (2001) Combining ability for nodulation in common bean (*Phaseolus vulgaris* L.) genotypes from Andean and middle American gene pools. Euphytica 118:265–270. https://doi.org/10.1023/a:1017560118666

Gardner CO, Eberhart SA (1966) Analysis and interpretation of the variety cross diallel and related populations. Biometrics 22:439–452. https://doi.org/10.2307/2528181

Gilmoure A, Gogel BJ, Cullis BR, Whelam SJ, Thompson R (2015) ASReml user guide release 4.1 structural specification. VSN International Ltd, Hemel Hempstead, HP1 1ES, UK

Griffing B (1956) Concept of general and specific combining ability in relation to diallel crossing systems. Aust J Biol Sci 9:463–493

Hadfield JD (2010) MCMC methods for multi-response generalized linear mixed models: the MCMCglmm R package. J Stat Softw 33(2):1–22. https://doi.org/10.18637/jss.v033.i02

Harriman J, Nwammadu C (2016) Utilization of diallel analyses for heritability, GCA and SCA studies in crop improvement. Am Adv J Biol Sci 2:159–167

Hayman BI (1954) The analysis of variance of diallel tables. Biometrics 10:235–244. https://doi.org/10.2307/3001877

Kery M (2010) Introduction to WinBUGS for ecologists. A Bayesian approach to regression, ANOVA, mixed models and related analyses. Academic Press, Burlington, MA (USA)

Lenarcic AB, Svenson KL, Churchill GA, Valdar W (2012) A general bayesian approach to analyzing diallel crosses of inbred strains. Genetics 190:413–435. https://doi.org/10.1534/genetics.111.132563

Li H, Loken E (2002) A unified theory of statistical analysis and inference for variance component models for dyadic data. Stat Sin 12:519–535

Li Z, Coffey L, Garfin J, White MR, Spalding EP, de Leon N, Kaeppler SM, Schnable PS, Springer NM, Hirsch CN (2018) Genotype-by-environment interactions affecting heterosis in maize. PLoS ONE 13:e0191321. https://doi.org/10.1371/journal.pone.0191321

Lv A-Z, Zhang H, Zhang Z-X et al (2012) Conversion of the statistical combining ability into a genetic concept. J Integr Agric 11:43–52

Mahgoub GMA (2011) Partitioning of general and specific combining ability effects for estimating maternal and reciprocal effects. J Agric Sci 3:213–222. https://doi.org/10.5539/jas.v3n2p213

Makumbi D, Alvarado G, Crossa J, Burgueño J (2018) SASHAYDIALL: a SAS program for Hayman's diallel analysis. Crop Sci 58:1605–1615. https://doi.org/10.2135/cropsci2018.01.0047

Möhring J, Melchinger AE, Piepho HP (2011) REML-based diallel analysis. Crop Sci 51:470–478. https://doi.org/10.2135/cropsci2010.05.0272

Murray LW, Ray IM, Dong H, Segovia-Lerma A (2003) Clarification and reevaluation of population-based diallel analyses. Crop Sci 43:1930–1937. https://doi.org/10.2135/cropsci2003.1930

Napolitano M, Terzaroli N, Kashyap S, Russi L, Jones-Evans E, Albertini E (2020) Exploring heterosis in melon (*Cucumis melo* L.). Plants 9:282. https://doi.org/10.3390/plants9020282

Onofri A (2015) Confidence intervals: Am I unconsciously Bayesian? Commun Biom Crop Sci 10:58–64

Piepho HP, Emrich K (2005) Simultaneous confidence intervals for two estimable functions and their ratio under a linear model. Am Stat 59:292–300. https://doi.org/10.1198/000313005X70605

Piepho HP, Möhring J, Melchinger AE, Buchse A (2008) BLUP for phenotypic selection in plant breeding and variety testing. Euphytica 161:209–228. https://doi.org/10.1007/s10681-007-9449-8

Pinheiro JC, Bates DM (2000) Mixed-effects models in S and S-PLUS. Springer-Verlag, New York. https://doi.org/10.1007/b98882

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2020) nlme: linear and nonlinear mixed effects models. R package version 3.1–148, https://CRAN.R-project.org/package=nlme

Plummer M (2019) rjags: bayesian graphical models using MCMC. R package version 4–10. Available at: https://CRAN.R-project.org/package=rjags. Date of last access: 09 July 2020

Quimio CA, Zapata FJ (2019) Diallel analysis of callus induction and green-plant regeneration in rice anther culture. Crop Sci 30:188–192

R Core Team (2019) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria

Singh KB, Jain RP (1971) Analysis of diallel cross in *Phaseolus aureus* roxb. Theor Appl Genet 41:279–281. https://doi.org/10.1007/BF00277798

Singh I, Paroda R, Behl R (1986) Diallel analysis for combining ability over environments in wheat. Wheat Inf Serv 61–62:74–76

Singh M, Gupta S, Parsad R (2015) Design and analysis of experiments, vol 3. In: Hinkelmann K (ed) Special designs and applications. Wiley, New Jersey, USA, pp 1–69

Spiegelhalter D, Thomas A, Best N, Lunn D (2003) WinBUGS user manual, version 1.4. MRC Biostatistics Unit, Cambridge, UK

Sprague GF, Tatum LA (1942) General versus specific combining ability in single crosses of corn. Agron J 34:923–932. https://doi.org/10.2134/agronj1942.00021962003400100008x

Tong C, Liu G, Yang L, Shi J (2012) GSCA: new software and algorithms to analyse diallel mating designs based on restricted linear model. Silvae Genet. https://doi.org/10.1515/sg-2012-0016

Turner SD, Maurizio PL, Valdar W, Yandell BS, Simon PW (2018) Dissecting the genetic architecture of shoot growth in carrot (*Daucus carota* L.) using a diallel mating design. G3 8:411–426

Weisberg S (2005) Applied linear regression, 3rd edn. Wiley, Hoboken. https://doi.org/10.1002/0471704091

Wickham H (2011) The split-apply-combine strategy for data analysis. J Stat Softw 40:1–29. https://doi.org/10.18637/jss.v040.i01

Wu HX, Matheson AC (2000) Analysis of half-diallel mating design with missing crosses: theory and SAS program for testing and estimating GCA and SCA fixed effects. Silvae Genet 49:130–137

Wu H, Matheson AC (2001) Analyses of half-diallel mating designs with missing crosses: theory and SAS program for testing and estimating GCA and SCA variance components. Silvae Genet 50:265–271

Xiang B, Li B (2001) A new mixed analytical method for genetic analysis of diallel data. Can J For Res 31:2252–2259. https://doi.org/10.1139/x01-154

Xu ZC, Zhu J (1999) An approach for predicting heterosis based on an additive, dominance and additive × additive model with environment interaction. Heredity 82:510–517. https://doi.org/10.1038/sj.hdy.6884800

Yao WH, Zhang YD, Kang MS, Chen HM, Liu L, Yu LJ, Fan XM (2013) Diallel analysis models: a comparison of certain genetic statistics. Crop Sci 53:1481–1490. https://doi.org/10.2135/cropsci2013.01.0027

Yaseen M, Eskridge KM (2020) DiallelAnalysisR: diallel analysis with R.R package version 0.3.0, https://cran.r-project.org/package=DiallelAnalysisR

Zhang Y, Kang MS (1997) DIALLEL-SAS: a SAS program for Griffing's diallel analyses. Agron J 89:176–182. https://doi.org/10.2134/agronj1997.00021962008900020005x

Zhang Y, Kang MS, Lamkey KR (2005) DIALLEL-SAS05: a comprehensive program for Griffing's and Gardner–Eberhart analyses. Agron J 97:1097–1106. https://doi.org/10.2134/agronj2004.0260

# Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com