In this post we are going to talk about the Gardner-Eberarth models, which are particularly suitable to describe heterotic effects. The peculiar trait of these models is that they consider different means for crosses and selfed parents and, therefore, they are reserved for the mating designs 2 (selfed parents and crosses, but no reciprocals) or 1 (selfed parents, crosses and reciprocals). The first model is know as GE2 model and it is specified as:

$$y_{ijk} = \mu_{\nu} + \gamma_k + 0.5 \, \left(v_i + v_j \right) + \bar{h} + h_i + h_j + s_{ij} + \varepsilon_{ijk} \quad\quad\quad (1)$$

where γ_k is the effect of block k and μ_{ν} is the overall mean for all selfed parents (not the overall mean, as in other diallel 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 (μ_{ν}). According to the authors, this would be the Variety Effect (VE); as a consequence, the expected value for the ith selfed parent is $\mu_{\nu} + v_i$, while the expected value for the cross ij, in absence of any dominance/heterosis effects, would be $\mu_{\nu} + 0.5 \, \left(v_i + v_j \right)$, that is the mean value of its parents. There is a close relationship between g_i and g_j in Griffing's equations ([see here](#)) and v_i and v_j in the GE2 equation (Eq. 1), that is: $v_i = 2\, g_i + (n - 2) d_i$; therefore, the sum of squares for the GCA and VE effects are the same, although the estimates are different.

Since a cross does not necessarily respond according to the mean value of its parents, the parameter $\bar{h}$ represents the average heterosis (H.BAR) contributed by the 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 ith parent in its crosses (Hi), while s_{ij} is the Specific Combining Ability (SCA) for the cross between the ith and j^{th} parents, that is totally equivalent to the corresponding parameter in Griffing's models.

It is clear that both the Hayman's model type 2 and the GE2 model account for the heterosis effect, although they do it in a different way: in Hayman's model type 2 the specific effect of heterosis is assessed with reference to the overall mean, while in GE2 it is assessed by comparing the mean of a cross with the means of its parents. Indeed, the sum of squares for the 'MDD' effect in Hayman's model and 'Hi' effect in GE2 model 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:

$$\left\{ {\begin{array}{ll} y_{ijk} = \mu_{\nu} + \gamma_k + \bar{h} + \textrm{gc}_i + \textrm{gc}_j + s_{ij} & \textrm{if} \quad i \neq j\\ y_{ijk} = \mu_{\nu} + \gamma_k + \textrm{sp}_i & \textrm{if} \quad i = j \end{array}} \right. \quad\quad\quad (2)$$

Equation 2 is an array of two separate elements for crosses and selfed parents. For the crosses (equation above), the parameters $\textrm{gc}_i$ and $\textrm{gc}_j$ represent the GCA for the i and j parents in all their crosses (GCAC); it should be noted that GCA $\neq$ GCAC, as this latter effect is estimated without considering the selfed parents. The parameters $s{ij}$ are the same as in the previous models (Hayman's and Griffing's models: SCA effect), while $\textrm{sp}_i$ represent the effects of selfed parents (SP): they are numerically equivalent to the corresponding effects in Equation 1, but the sum of squares are different (see Murray et al., 2003). Therefore, we use different names for these two effects (SP and Hi).

Example 1: a half-diallel (no reciprocals)

As an example of a diallel experiments with no reciprocals, we consider the data reported in Lonnquist and Gardner (1961) relating to the yield of 21 maize genotypes, obtained from six male and six female parentals. The dataset is available as `lonnquist61` in the `lmDiallel` package; in the box below we load the data, after installing (if necessary) and loading the 'lmDiallel' package.

```
# library(devtools) # Install if necessary
# install_github("OnofriAndreaPG/lmDiallel")
library(lmDiallel)
library(multcomp)
data("lonnquist61")
```

For this complete diallel experiment we can fit equation 1 (model GE2), by including the functions `H.BAR()`, `VEi()`, `Hi()` and `SCA()`; we can use either the `lm()` or the `lm.diallel()` functions, as shown in the box below.

```
dMod <- lm(Yield ~ H.BAR(Par1, Par2) + VEi(Par1, Par2) +
              Hi(Par1, Par2) + SCA(Par1, Par2),
           data = lonnquist61)
dMod2 <- lm.diallel(Yield ~ Par1 + Par2,
                    data = lonnquist61, fct = "GE2")
```

In this case the dataset has no replicates and, therefore, we need to provide an estimate of the residual mean square and degrees of freedom. If we have fitted the model by using the `lm()` function, the resulting 'lm' object can be explored by using the `summary.diallel()` and `anova.diallel()` functions. Otherwise, if we have fitted the model with the `lm.diallel()` function, the resulting 'diallel' object can be explored by using the `summary()` and `anova()` methods. See the box below for an example: the results are, obviously, the same.

```
# summary.diallel(dMod, MSE = 7.1, dfr = 60)
anova.diallel(dMod, MSE = 7.1, 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
# summary(dMod2, MSE = 7.1, dfr = 60)
anova(dMod2, MSE = 7.1, dfr = 60)
## Analysis of Variance Table
##
## Response: Yield
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## h.bar       1 115.440 115.440 16.2592 0.0001583 ***
## Variety     5 234.230  46.846  6.5980 5.923e-05 ***
## h.i         5  59.720  11.944  1.6823 0.1527246
```

```
## SCA          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
```

Also for the diallel object, we can retrieve the full list of genetical parameters with the `glht()` function, by using the same syntax as shown above.

```
gh <- glht(linfct = diallel.eff(dMod2, MSE = 7.1, dfr = 60))
summary(gh, test = adjusted(type = "none"))
#    Simultaneous Tests for General Linear Hypotheses
#
# Linear Hypotheses:
#                Estimate Std. Error t value Pr(>|t|)
# Intercept == 0    92.450     1.088  84.987  < 2e-16 ***
# h.bar == 0         5.190     1.287   4.032  0.00043 ***
# v_B == 0           4.150     2.432   1.706  0.09991 .
# v_G == 0          -4.550     2.432  -1.871  0.07270 .
# v_H == 0          -0.750     2.432  -0.308  0.76028
# v_K == 0          -1.150     2.432  -0.473  0.64031
# v_K2 == 0          3.750     2.432   1.542  0.13524
# v_M == 0          -1.450     2.432  -0.596  0.55625
#...
#...
# s_K2:K == 0        0.585     2.064   0.283  0.77909
# s_K2:M == 0       -1.115     2.064  -0.540  0.59364
# s_M:B == 0        -1.040     2.064  -0.504  0.61859
# s_M:G == 0        -2.290     2.064  -1.110  0.27737
# s_M:H == 0         3.385     2.064   1.640  0.11304
# s_M:K == 0         1.060     2.064   0.514  0.61189
# s_M:K2 == 0       -1.115     2.064  -0.540  0.59364
```

If we want to fit Equation 2 (model GE3), we can follow a very similar approach, by using the functions `H.BAR()`, `SP()`, `GCAC()` and `SCA()`. The box below shows an example either with the `lm()` or the with the `lm.diallel()` functions.

```
dMod <- lm(Yield ~ H.BAR(Par1, Par2) + SP(Par1, Par2)
           + GCAC(Par1, Par2) + SCA(Par1, Par2),
           data = lonnquist61)
dMod2 <- lm.diallel(Yield ~ Par1 + Par2,
                    data = lonnquist61, fct = "GE3")

# summary.diallel(dMod, MSE = 7.1, dfr = 60)
anova.diallel(dMod, MSE = 7.1, 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 ***
## SP(Par1, Par2)     5  55.975  11.195  1.5768 0.1804080
## GCAC(Par1, Par2)   5 237.975  47.595  6.7035 5.069e-05 ***
## 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
# summary(dMod2, MSE = 7.1, dfr = 60)
anova(dMod2, MSE = 7.1, dfr = 60)
## Analysis of Variance Table
##
## Response: Yield
##               Df  Sum Sq Mean Sq F value     Pr(>F)
## h.bar          1 115.440 115.440 16.2592 0.0001583 ***
## Selfed par.    5  55.975  11.195  1.5768 0.1804080
## Varieties      5 237.975  47.595  6.7035 5.069e-05 ***
## SCA            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
```

Also for the diallel object, we can retrieve the full list of genetical parameters with the `glht()` function, by using the same syntax as shown above.

```
gh <- glht(linfct = diallel.eff(dMod2, MSE = 7.1, dfr = 60))
summary(gh, test = adjusted(type = "none"))
##
##    Simultaneous Tests for General Linear Hypotheses
##
## Linear Hypotheses:
##                  Estimate Std. Error t value Pr(>|t|)
## Intercept == 0     92.450      1.088  84.987  < 2e-16 ***
## h.bar == 0          5.190      1.287   4.032  0.00043 ***
## sp_B == 0           4.150      2.432   1.706  0.09991 .
## sp_G == 0          -4.550      2.432  -1.871  0.07270 .
## sp_H == 0          -0.750      2.432  -0.308  0.76028
## sp_K == 0          -1.150      2.432  -0.473  0.64031
## sp_K2 == 0          3.750      2.432   1.542  0.13524
## sp_M == 0          -1.450      2.432  -0.596  0.55625
## gc_B == 0           0.900      1.216   0.740  0.46593
## gc_G == 0          -2.050      1.216  -1.686  0.10385
## gc_H == 0          -0.025      1.216  -0.021  0.98376
## gc_K == 0          -3.000      1.216  -2.467  0.02055 *
## gc_K2 == 0          6.375      1.216   5.242 1.78e-05 ***
## gc_M == 0          -2.200      1.216  -1.809  0.08205 .
## s_B:G == 0          4.810      2.064   2.330  0.02781 *
## s_B:H == 0         -1.415      2.064  -0.686  0.49905
## s_B:K == 0         -0.140      2.064  -0.068  0.94644
## s_B:K2 == 0        -2.215      2.064  -1.073  0.29305
## s_B:M == 0         -1.040      2.064  -0.504  0.61859
## s_G:B == 0          4.810      2.064   2.330  0.02781 *
## s_G:H == 0         -2.865      2.064  -1.388  0.17689
## s_G:K == 0         -0.990      2.064  -0.480  0.63548
## s_G:K2 == 0         1.335      2.064   0.647  0.52342
## s_G:M == 0         -2.290      2.064  -1.110  0.27737
## s_H:B == 0         -1.415      2.064  -0.686  0.49905
## s_H:G == 0         -2.865      2.064  -1.388  0.17689
```

```
## s_H:K == 0         -0.515     2.064   -0.250   0.80492
## s_H:K2 == 0          1.410     2.064    0.683   0.50056
## s_H:M == 0           3.385     2.064    1.640   0.11304
## s_K:B == 0          -0.140     2.064   -0.068   0.94644
## s_K:G == 0          -0.990     2.064   -0.480   0.63548
## s_K:H == 0          -0.515     2.064   -0.250   0.80492
## s_K:K2 == 0          0.585     2.064    0.283   0.77909
## s_K:M == 0           1.060     2.064    0.514   0.61189
## s_K2:B == 0         -2.215     2.064   -1.073   0.29305
## s_K2:G == 0          1.335     2.064    0.647   0.52342
## s_K2:H == 0          1.410     2.064    0.683   0.50056
## s_K2:K == 0          0.585     2.064    0.283   0.77909
## s_K2:M == 0         -1.115     2.064   -0.540   0.59364
## s_M:B == 0          -1.040     2.064   -0.504   0.61859
## s_M:G == 0          -2.290     2.064   -1.110   0.27737
## s_M:H == 0           3.385     2.064    1.640   0.11304
## s_M:K == 0           1.060     2.064    0.514   0.61189
## s_M:K2 == 0         -1.115     2.064   -0.540   0.59364
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- none method)
#      Simultaneous Tests for General Linear Hypotheses
#
# Linear Hypotheses:
#                 Estimate Std. Error t value Pr(>|t|)
# Intercept == 0   92.450      1.088  84.987  < 2e-16 ***
# h.bar == 0        5.190      1.287   4.032  0.00043 ***
# sp_B == 0         4.150      2.432   1.706  0.09991 .
# sp_G == 0        -4.550      2.432  -1.871  0.07270 .
# sp_H == 0        -0.750      2.432  -0.308  0.76028
# sp_K == 0        -1.150      2.432  -0.473  0.64031
# sp_K2 == 0        3.750      2.432   1.542  0.13524
# ...
# ...
# s_K2:H == 0       1.410      2.064   0.683  0.50056
# s_K2:K == 0       0.585      2.064   0.283  0.77909
# s_K2:M == 0      -1.115      2.064  -0.540  0.59364
# s_M:B == 0       -1.040      2.064  -0.504  0.61859
# s_M:G == 0       -2.290      2.064  -1.110  0.27737
# s_M:H == 0        3.385      2.064   1.640  0.11304
# s_M:K == 0        1.060      2.064   0.514  0.61189
# s_M:K2 == 0      -1.115      2.064  -0.540  0.59364
```

Example 2: a full diallel experiment

If we have a full diallel experiment (with reciprocals), we can fit Equations 1 and 2, but we should also include the reciprocal effects, in order to avoid that the residual term is inflated and no longer provides a reliable estimate of the experimental error. We provide an example with the data in Hayman (1954), relating to a complete diallel experiment with eight parental lines, producing 64 combinations (8 selfs + 28 crosses with 2 reciprocals each). The R dataset is included in the 'lmDiallel' package and the models are fitted by using the same coding as above, apart from the fact that the function REC() is included in the lm() call and the arguments

"GE2r" and "GE3r" are used instead of "GE2" and "GE3" in the `lm.diallel()` call.

```
data("hayman54")
contrasts(hayman54$Block) <- "contr.sum"
dMod <- lm(Ftime ~ Block + H.BAR(Par1, Par2) + VEi(Par1, Par2) +
              Hi(Par1, Par2) + SCA(Par1, Par2) + REC(Par1, Par2),
           data = hayman54)
dMod2 <- lm.diallel(Ftime ~ Par1 + Par2, Block = Block,
                    data = hayman54, fct = "GE2r")
# summary(dMod2)
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
## h.bar         1  30797   30797 73.8840 3.259e-12 ***
## Variety       7 277717   39674 95.1805 < 2.2e-16 ***
## h.i           7  34153    4879 11.7050 1.957e-09 ***
## SCA          20  37289    1864  4.4729 2.560e-06 ***
## Reciprocals  28  19112     683  1.6375   0.05369 .
## Residuals    63  26260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
gh <- glht(linfct = diallel.eff(dMod2))
summary(gh, test = adjusted(type = "none"))
#    Simultaneous Tests for General Linear Hypotheses
#
# Linear Hypotheses:
#                 Estimate Std. Error t value Pr(>|t|)
# Intercept == 0  2.039e+02  5.104e+00  39.956  < 2e-16 ***
# h.bar == 0     -4.690e+01  5.456e+00  -8.596 4.48e-09 ***
# v_A == 0        8.506e+01  1.350e+01   6.299 1.14e-06 ***
# v_B == 0       -3.344e+01  1.350e+01  -2.476 0.020115 *
# v_C == 0        1.841e+02  1.350e+01  13.630 2.37e-13 ***
# v_D == 0        3.706e+01  1.350e+01   2.745 0.010839 *
# v_E == 0       -3.794e+01  1.350e+01  -2.809 0.009301 **
# v_F == 0       -3.394e+01  1.350e+01  -2.513 0.018499 *
# v_G == 0       -1.509e+02  1.350e+01 -11.177 1.99e-11 ***
# v_H == 0       -4.994e+01  1.350e+01  -3.698 0.001023 **
# h_A == 0        4.885e+00  7.797e+00   0.627 0.536380
# ...
# ...
# r_H:C == 0     -5.500e+00  1.021e+01  -0.539 0.594620
# r_H:D == 0     -5.000e+00  1.021e+01  -0.490 0.628380
# r_H:E == 0     -8.500e+00  1.021e+01  -0.833 0.412617
# r_H:F == 0     -1.750e+01  1.021e+01  -1.714 0.098370 .
# r_H:G == 0     -1.400e+01  1.021e+01  -1.371 0.181956
```

The code for the GE3 model with reciprocal effects is shown in the box below.

```
dMod <- lm(Ftime ~ Block + H.BAR(Par1, Par2) + SP(Par1, Par2)
```

```
                + GCAC(Par1, Par2) + SCA(Par1, Par2) + REC(Par1, Par2),
             data = hayman54)
dMod2 <- lm.diallel(Ftime ~ Par1 + Par2, Block = Block,
                    data = hayman54, fct = "GE3r")
# summary(dMod2)
anova(dMod2)
## Analysis of Variance Table
##
## Response: Ftime
##               Df Sum Sq Mean Sq F value    Pr(>F)
## Block          1    142   142.4  0.3416   0.56100
## h.bar          1  30797 30796.9 73.8840 3.259e-12 ***
## gcac           7 168923 24131.9 57.8941 < 2.2e-16 ***
## Selfed par.    7 142946 20420.9 48.9913 < 2.2e-16 ***
## SCA           20  37289  1864.4  4.4729 2.560e-06 ***
## Reciprocals   28  19112   682.6  1.6375   0.05369 .
## Residuals     63  26260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# anova(dMod)
gh <- glht(linfct = diallel.eff(dMod2))
# summary(gh, test = adjusted(type = "none"))
```

Estimation of variance components (random genetic effects)

If we intend to regard the genetic effects as random and to estimate variance components, we can use the `mmer()` function in the 'sommer' package (Covarrubias-Pazaran, 2016), although we need to code a bunch of dummy variables. In order to make things simpler for routine experiments, we have coded the `mmer.diallel()` wrapper using the same syntax as the `lm.diallel()` function. The exemplary code is given in the box below.

```
# Random genetic effects
mod1m <- mmer.diallel(Yield ~ Par1 + Par2,
                      data = lonnquist61,
                      fct = "GE2")
mod1m
##          VarComp VarCompSE
## Variety 2.344044  2.333869
## h.i     5.172099  4.905691
## SCA     6.142047  2.789668
mod2m <- mmer.diallel(Yield ~ Par1 + Par2,
                      data = lonnquist61,
                      fct = "GE3")
mod2m
##               VarComp VarCompSE
## GCAC         10.125567  7.563026
## Selfed par.   4.107823  7.830039
## SCA           7.087220  3.342822
mod3m <- mmer.diallel(Ftime ~ Par1 + Par2,
                      data = hayman54,
                      fct = "GE2r")
```

```
mod3m
##               VarComp   VarCompSE
## Variety     2347.35935 1279.94018
## h.i          634.70067  408.56286
## SCA          362.24772  148.53288
## Reciprocals   66.78085   49.16288
## Residuals    415.44775   73.43871
mod4m <- mmer.diallel(Ftime ~ Par1 + Par2,
                      data = hayman54,
                      fct = "GE3r")
mod4m
##                 VarComp   VarCompSE
## GCAC            927.78740  537.89968
## Selfed par. 10003.93261 5456.47108
## SCA            362.96912  148.50097
## Reciprocals     67.50895   49.11942
## Residuals      412.54141   72.93144
```

Thanks for reading!