Quantitative genetics using the sommer package

Giovanny Covarrubias-Pazaran

2018-03-03

The sommer package was developed to provide R users a powerful and reliable multivariate mixed model solver for different genetic and non-genetic analysis in diploid and polyploid organisms. This package allows the user to estimate variance components for a mixed model with the advantage of specifying the variance-covariance structure of the random effects, specify heterogeneous variances, and obtain other parameters such as BLUPs, BLUEs, residuals, fitted values, variances for fixed and random effects, etc.

The package is focused on problems of the type p > n related to genomic prediction (hybrid prediction & genomic selection) and GWAS analysis, although any general mixed model can be fitted as well. The package provides kernels to estimate additive (`A.mat`), dominance (`D.mat`), and epistatic (`E.mat`) relationship matrices that have been shown to increase prediction accuracy under certain scenarios or simply to estimate the variance components of such. The package provides flexibility to fit other genetic models such as full and half diallel models as well.

Vignettes aim to provide several examples in how to use the sommer package under different scenarios. We will spend the rest of the space providing examples for:

1) Heritability (h^2) calculation
2) Specifying heterogeneous variances in mixed models
3) Using the pin calculator
4) Half and full diallel designs (using the overlay)
5) Genomic selection (predicting mendelian sampling)
6) Single cross prediction (hybrid prediction)
7) Spatial modeling (using the 2-dimensional splines)
8) Multivariate genetic models and genetic correlations

Background

The core of the package are the `mmer2` (formula-based) and `mmer` (matrix-based) functions which solve the mixed model equations. The functions are an interface to call the `NR` Direct-Inversion Newton-Raphson (Tunnicliffe 1989; Gilmour et al. 1995; Lee et al. 2016) or the `EMMA` efficient mixed model association algorithm (Kang et al. 2008). Since version 2.0 sommer can handle multivariate models. Following Maier et al. (2015), the multivariate (and by extension the univariate) mixed model implemented has the form:

$y_1 = X_1\beta_1 + Z_1u_1 + \epsilon_1$ $y_2 = X_2\beta_2 + Z_2u_2 + \epsilon_2$ $\ldots$ $y_i = X_i\beta_i + Z_iu_i + \epsilon_i$

where y_i is a vector of trait phenotypes, β_i is a vector of fixed effects, u_i is a vector of random effects for individuals and e_i are residuals for trait 'i' (i = 1, …, t). The random effects (u_1 … u_i and e_i) are assumed to be normally distributed with mean zero. X and Z are incidence matrices for fixed and random effects respectively. The distribution of the multivariate response and the phenotypic variance covariance (V) are:

$Y = X\beta + ZU + \epsilon_i$

Y ~ MVN($X\beta$, V)

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_t \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} X_1 & ... & ... \\ \vdots & \ddots & \vdots \\ ... & ... & X_t \end{bmatrix}$$

$$\mathbf{V} = \begin{bmatrix} Z_1 K \sigma^2_{g_1} Z_1' + Z_1 I \sigma^2_{\epsilon_1} Z_1' & ... & Z_1 K \sigma_{g_{1,t}} Z_t' + Z_1 I \sigma_{\epsilon_{1,t}} Z_t' \\ \vdots & \ddots & \vdots \\ Z_1 K \sigma_{g_{1,t}} Z_t' + Z_1 I \sigma_{\epsilon_{1,t}} Z_t' & ... & Z_t K \sigma^2_{g_t} Z_t' + Z_t I \sigma^2_{\epsilon_t} Z_t' \end{bmatrix}$$

where K is the relationship or covariance matrix for the kth random effect (u=1,. . . ,k), and R=I is an identity matrix for the residual term. The terms $\sigma^2_{g_i}$ and $\sigma^2_{\epsilon_i}$ denote the genetic (or any of the kth random terms) and residual variance of trait 'i', respectively and $\sigma_{g_{ij}}$ and $\sigma_{\epsilon_{ij}}$ the genetic (or any of the kth random terms) and residual covariance between traits 'i' and 'j' (i=1,. . . ,t, and j=1,. . . ,t). The algorithm implemented optimizes the log likelihood:

$logL = 1/2 * ln(|V|) + ln(X'|V|X) + Y'PY$

where || is the determinant of a matrix. And the REML estimates are updated using a Newton optimization algorithm of the form:

$\theta^{k+1} = \theta^k + (H^k)^{-1} * \frac{dL}{d\sigma^2_i} | \theta^k$

Where, θ is the vector of variance components for random effects and covariance components among traits, H^{-1} is the inverse of the Hessian matrix of second derivatives for the kth cycle, $\frac{dL}{d\sigma^2_i}$ is the vector of first derivatives of the likelihood with respect to the variance-covariance components. The Eigen decomposition of the relationship matrix proposed by Lee and Van Der Werf (2016) was included in the Newton-Raphson algorithm to improve time efficiency. Additionally, the popular pin function to estimate standard errors for linear combinations of variance components (i.e. heritabilities and genetic correlations) was added to the package as well.

The function `mmer` takes the Zs and Ks for each random effect and construct the neccesary structure inside and estimates the variance components by ML/REML using any of the 4 methods available in sommer. The `mmer2` function is enabled to work in a model-based fashion so user don't have to build the Z's and K matrices. Please refer to the canonical papers listed in the Literature section to check how the algorithms work. We have tested widely the methods to make sure they provide the same solution when the likelihood behaves well but for complex problems they might lead to slightly different answers. If you have any concern please contact me at cova_ruber@live.com.mx.

In the following section we will go in detail over several examples on how to use mixed models in univariate and multivariate case and their use in quantitative genetics.

1) Marker and non-marker based heritability calculation

The heritability is one of the most popular parameters among the breeding and genetics community because of the insight that provides in the inheritance of the trait. The heritability is usually estimated as narrow sense (h^2; only additive variance in the numerator σ^2_A), and broad sense (H^2; all genetic variance in the numerator σ^2_G).

In a classical breeding experiment with no molecular markers, special designs are performed to estimate and disect the additive (σ^2_A) and non-additive (i.e. dominance σ^2_D) variance along with environmental variability. Designs such as generation analysis, North Carolina designs are used to disect σ^2_A and σ^2_D to estimate the narrow sense heritability (h^2). When no special design is available we can still disect the genetic variance (σ^2_G) and estimate the broad sense heritability. In this first example we will show the broad sense estimation which doesn't use covariance structures for the genotipic effect (i.e. genomic or additive relationship matrices). For big models with no covariance structures, sommer's direct inversion is a bad idea to use but we will show anyways how to do it, but keep in mind that for very sparse models we recommend using the lmer function from the lme4 package or any other package using MME-based algorithms (i.e. asreml-R).

The following dataset has 41 potato lines evaluated in 5 locations across 3 years in an RCBD design. We show how to fit the model and extract the variance components to calculate the h^2.

```
library(sommer)
data(h2example)
head(h2example)
```

```
##                  Name     Env Loc Year     Block y
## 1            W8822-3 FL.2012  FL 2012 FL.2012.1 2
## 2            W8867-7 FL.2012  FL 2012 FL.2012.2 2
## 3           MSL007-B MO.2011  MO 2011 MO.2011.1 3
## 4         CO00270-7W FL.2012  FL 2012 FL.2012.2 3
## 5 Manistee(MSL292-A) FL.2013  FL 2013 FL.2013.2 3
## 6           MSM246-B FL.2012  FL 2012 FL.2012.2 3
```

```
ans1 <- mmer2(y~1,
              random = ~Name + Env + Name:Env + Block,
              rcov = ~units,
              data=h2example, silent = TRUE)

suma <- summary(ans1)
n.env <- length(levels(h2example$Env))
pin(ans1, h2 ~ V1 / ( V1 + (V3/n.env) + (V5/(2*n.env)) ) )
```

```
##     Estimate         SE
## h2 0.8594794 0.03517549
```

The same model can be fitted with the `mmer` function that is actually used by the `mmer2` function in the background. This is just to show that you can create your customized matrices and use the mixed model solver. This is how you would do it:

```
library(sommer)
data(h2example)
head(h2example)
```

```
##                  Name     Env Loc Year     Block y
## 1            W8822-3 FL.2012  FL 2012 FL.2012.1 2
## 2            W8867-7 FL.2012  FL 2012 FL.2012.2 2
## 3           MSL007-B MO.2011  MO 2011 MO.2011.1 3
## 4         CO00270-7W FL.2012  FL 2012 FL.2012.2 3
## 5 Manistee(MSL292-A) FL.2013  FL 2013 FL.2013.2 3
## 6           MSM246-B FL.2012  FL 2012 FL.2012.2 3
```

```
Z1 <- model.matrix(~Name-1, h2example)
Z2 <- model.matrix(~Env-1, h2example)
Z3 <- model.matrix(~Env:Name-1, h2example)
Z4 <- model.matrix(~Block-1, h2example)
ETA <- list(name=list(Z=Z1),env=list(Z=Z2),name.env=list(Z=Z3),block=list(Z=Z4))
y <- h2example$y
ans1 <- mmer(Y=y, Z=ETA, silent = TRUE)
vc <- ans1$var.comp
```

Recently with markers becoming cheaper, thousand of markers can be run in the breeding materials. When markers are available, an special design is not neccesary to disect the additive genetic variance. The availability of the additive, dominance and epistatic relationship matrices allow us to estimate σ^2_A, σ^2_D and σ^2_I, although given that A, D and E are not orthogonal the interpretation of models that fit more than A and D become cumbersome.

Assume you have a population (even unreplicated) in the field but in addition we have genetic markers. Now we can fit the model and estimate the genomic heritability that explains a portion of the additive genetic variance (with high marker density $\sigma^2_A = \sigma^2_g$)

```
data(CPdata)
CPpheno$idd <-CPpheno$id; CPpheno$ide <-CPpheno$id
### look at the data
head(CPpheno)
```

```
##        id Row Col Year      color  Yield FruitAver Firmness Rowf Colf  idd
## P003 P003   3   1 2014 0.10075269 154.67     41.93  588.917    3    1 P003
## P004 P004   4   1 2014 0.13891940 186.77     58.79  640.031    4    1 P004
## P005 P005   5   1 2014 0.08681502  80.21     48.16  671.523    5    1 P005
## P006 P006   6   1 2014 0.13408561 202.96     48.24  687.172    6    1 P006
## P007 P007   7   1 2014 0.13519278 174.74     45.83  601.322    7    1 P007
## P008 P008   8   1 2014 0.17406685 194.16     44.63  656.379    8    1 P008
##       ide
## P003 P003
## P004 P004
## P005 P005
## P006 P006
## P007 P007
## P008 P008
```

```
CPgeno[1:5,1:4]
```

```
##      scaffold_50439_2381 scaffold_39344_153 uneak_3436043 uneak_2632033
## P003                   0                  0             0             1
## P004                   0                  0             0             1
## P005                   0                 -1             0             1
## P006                  -1                 -1            -1             0
## P007                   0                  0             0             1
```

```
## fit a model including additive and dominance effects
A <- A.mat(CPgeno) # additive relationship matrix
D <- D.mat(CPgeno) # dominance relationship matrix
E <- E.mat(CPgeno) # epistatic relationship matrix

ans.ADE <- mmer2(color~1,
                 random=~g(id) + g(idd) + g(ide),
                 rcov=~units,
                 G=list(id=A,idd=D,ide=E),
                 silent = TRUE, data=CPpheno)
suma <- summary(ans.ADE)$var.comp.table
(H2 <- sum(suma[1:3,1])/sum(suma[,1]))
```

```
## [1] 0.7224969
```

```
(h2 <- sum(suma[1,1])/sum(suma[,1]))
```

```
## [1] 0.4827271
```

In the previous example we showed how to estimate the additive (σ^2_A), dominance (σ^2_D), and epistatic (σ^2_I) variance components based on markers and estimate broad (H^2) and narrow sense heritability (h^2). Notice that we used the `g()` function which indicates that the random effect inside the parenthesis (i.e. id, idd or ide) has a covariance matrix (A, D, or E), that will be specified in the G argument in the form of a list and using the name of the random efect to allow the program to recognize which variance covariance matrix belongs to each random effect. Please DO NOT provide the inverse but the original covariance matrix. This is why we have called the function `g()` and no `giv()` as the popular software asreml.

Just to show one more time that you can use your own matrices we will repeat the same calculation using the `mmer` function:

```
data(CPdata)
### look at the data
head(CPpheno)
```

```
##        id Row Col Year      color  Yield FruitAver Firmness Rowf Colf
## P003 P003   3   1 2014 0.10075269 154.67     41.93  588.917    3    1
## P004 P004   4   1 2014 0.13891940 186.77     58.79  640.031    4    1
## P005 P005   5   1 2014 0.08681502  80.21     48.16  671.523    5    1
## P006 P006   6   1 2014 0.13408561 202.96     48.24  687.172    6    1
## P007 P007   7   1 2014 0.13519278 174.74     45.83  601.322    7    1
## P008 P008   8   1 2014 0.17406685 194.16     44.63  656.379    8    1
```

```
CPgeno[1:5,1:4]
```

```
##      scaffold_50439_2381 scaffold_39344_153 uneak_3436043 uneak_2632033
## P003                   0                  0             0             1
## P004                   0                  0             0             1
## P005                   0                 -1             0             1
## P006                  -1                 -1            -1             0
## P007                   0                  0             0             1
```

```
## fit a model including additive and dominance effects
Z1 <- model.matrix(~id-1, CPpheno); colnames(Z1) <- gsub("id","",colnames(Z1))
A <- A.mat(CPgeno) # additive relationship matrix
D <- D.mat(CPgeno) # dominance relationship matrix
E <- E.mat(CPgeno) # epistatic relationship matrix
y <- CPpheno$color

ETA <- list(id=list(Z=Z1,K=A),idd=list(Z=Z1,K=D),ide=list(Z=Z1,K=E))
ans.ADE <- mmer(Y=y, Z=ETA, silent = TRUE)
ans.ADE$var.comp
```

```
## $id
##             T1
## T1 0.003668396
##
## $idd
##             T1
## T1 0.001820039
##
## $ide
##    T1
## T1  0
##
## $units
##             T1
## T1 0.002105616
```

2) Specifying heterogeneous variances in univariate models

Very often in multi-environment trials, the assumption that genetic variance is the same across locations may be too naive. Because of that, specifying a general genetic component and a location specific genetic variance is the way to go. Although the function 'mmer' implemented in sommer can be used to do that, can be quite cumbersome and messy to create the incidence and variance covariance matrices for fitting those models. For that reason the function 'mmer2' was added to the package to make such models easier to fit.

We estimate variance components for GCA_2 and SCA specifying the variance structure.

```
data(cornHybrid)
hybrid2 <- cornHybrid$hybrid # extract cross data
head(hybrid2)
```

```
##   Location GCA1   GCA2         SCA Yield PlantHeight
## 1        1 A258 AS5707 A258:AS5707    NA          NA
## 2        1 A258     B2     A258:B2    NA          NA
## 3        1 A258    B99    A258:B99    NA          NA
## 4        1 A258   LH51   A258:LH51    NA          NA
## 5        1 A258   Mo44   A258:Mo44    NA          NA
## 6        1 A258  NC320  A258:NC320    NA          NA
```

```
### fit the model
modFD <- mmer2(Yield~1,
               random=~ at(Location,c("3","4")):GCA2,
               rcov= ~ at(Location):units,
               data=hybrid2, silent = TRUE)
summary(modFD)
```

```
## ==================================================
##     Multivariate Linear Mixed Model fit by REML
## ******************  sommer 3.3  ******************
## ==================================================
##          logLik      AIC      BIC Method Converge
## Value -164.6839 331.3677 335.3592    MNR     TRUE
## ==================================================
## Variance-Covariance components:
##                     VarComp VarCompSE Zratio
## 3:GCA2.Yield-Yield    62.42     53.40  1.169
## 4:GCA2.Yield-Yield    98.02     79.59  1.232
## 1:units.Yield-Yield  216.82     30.76  7.048
## 2:units.Yield-Yield  216.82     30.76  7.048
## 3:units.Yield-Yield  493.07     77.29  6.380
## 4:units.Yield-Yield  711.98    111.64  6.378
## ==================================================
## Fixed effects:
##
## $Yield
##           Estimate Std. Error  t value
## Intercept 138.1129  0.9441655 146.2804
##
## ==================================================
## Groups and observations:
##        Observ Groups
## 3:GCA2    400     20
## 4:GCA2    400     20
## =====================================================
## Use the '$' sign to access results and parameters
```

In the previous example we showed how the `at` function is used in the `mmer2` solver. By using the `at` function you can specify that i.e. the GCA2 has a different variance in different Locations, in this case locations 3 and 4, but also a main GCA variance. This is considered a CS + DIAG (compound symmetry + diagonal) model.

In addition, other functions can be added on top to fit models with covariance structures, i.e. the `g()` function which indicates that the random effect inside the parenthesis (i.e. GCA2) has a covariance matrix (A, pedigree or genomic relationship matrix) that will be specified in the G argument in the form of a list:

```
data(cornHybrid)
hybrid2 <- cornHybrid$hybrid # extract cross data
## get the covariance structure for GCA2
A <- cornHybrid$K
## fit the model
modFD <- mmer2(Yield~1,
               random=~ g(GCA2) + at(Location):g(GCA2),
               rcov= ~ at(Location):units,
               data=hybrid2, G=list(GCA2=A),
               silent = TRUE, draw=FALSE)
summary(modFD)
```

```
## =====================================================
##     Multivariate Linear Mixed Model fit by REML
## ******************  sommer 3.3  ******************
## =====================================================
##          logLik      AIC      BIC Method Converge
## Value -157.4751 316.9502 320.9417    MNR     TRUE
## =====================================================
## Variance-Covariance components:
##                        VarComp VarCompSE Zratio
## g(GCA2).Yield-Yield     28.180     12.49 2.2566
## 1:g(GCA2).Yield-Yield    0.000       NaN 0.0000
## 2:g(GCA2).Yield-Yield    0.000       NaN 0.0000
## 3:g(GCA2).Yield-Yield    3.925     18.43 0.2130
## 4:g(GCA2).Yield-Yield   10.068     27.89 0.3611
## 1:units.Yield-Yield    187.925     29.07 6.4642
## 2:units.Yield-Yield    187.925     29.07 6.4642
## 3:units.Yield-Yield    497.124     76.36 6.5104
## 4:units.Yield-Yield    727.368    111.74 6.5093
## =====================================================
## Fixed effects:
##
## $Yield
##           Estimate Std. Error  t value
## Intercept 138.3383   1.321406 104.6903
##
## =====================================================
## Groups and observations:
##           Observ Groups
## g(GCA2)      400     20
## 1:g(GCA2)    400     20
## 2:g(GCA2)    400     20
## 3:g(GCA2)    400     20
## 4:g(GCA2)    400     20
## ==================================================
## Use the '$' sign to access results and parameters
```

The `draw` argument allows you to see the progress of the likelihood and the change of the variance components, we just mention it in case you like to do that inspection but this will make the fitting process more time consuming.

3) Using the pin calculator

Sometimes the user needs to calculate ratios or functions of specific variance-covariance components and obtain the standard error for such parameters. Examples of these are the genetic correlations, heritabilities, etc. Using the CPdata we will show how to estimate the heritability and the standard error using the pin function that uses the delta method to come up with these parameters. This can be extended for any linear combination of the variance components.

```
data(CPdata)
#### create the variance-covariance matrix
A <- A.mat(CPgeno)
#### look at the data and fit the model
head(CPpheno)
```

```
##        id Row Col Year      color  Yield FruitAver Firmness Rowf Colf
## P003 P003   3   1 2014 0.10075269 154.67     41.93  588.917    3    1
## P004 P004   4   1 2014 0.13891940 186.77     58.79  640.031    4    1
## P005 P005   5   1 2014 0.08681502  80.21     48.16  671.523    5    1
## P006 P006   6   1 2014 0.13408561 202.96     48.24  687.172    6    1
## P007 P007   7   1 2014 0.13519278 174.74     45.83  601.322    7    1
## P008 P008   8   1 2014 0.17406685 194.16     44.63  656.379    8    1
```

```
mix1 <- mmer2(color~1,
              random=~g(id),
              rcov=~units,
              G=list(id=A), data=CPpheno, silent=TRUE)
summary(mix1)
```

```
## ==================================================
##     Multivariate Linear Mixed Model fit by REML
## ******************  sommer 3.3  ******************
## ==================================================
##          logLik      AIC      BIC Method Converge
## Value -110.7406 223.4812 227.3728    MNR     TRUE
## ==================================================
## Variance-Covariance components:
##                      VarComp VarCompSE Zratio
## g(id).color-color 0.005123 0.0010395  4.929
## units.color-color 0.002743 0.0003002  9.137
## ==================================================
## Fixed effects:
##
## $color
##            Estimate  Std. Error  t value
## Intercept 0.1825652 0.002754956 66.26792
##
## ==================================================
## Groups and observations:
```

```
##       Observ Groups
## g(id)    362    363
## ====================================================
## Use the '$' sign to access results and parameters
```

```
#### run the pin function
pin(mix1, h2 ~ V1 / ( V1 + V2 ) )
```

```
##     Estimate         SE
## h2 0.6512726 0.06109097
```

The same can be used for multivariate models. Please check the documentation of the `pin` function to see more examples.

4) Half and full diallel designs (use of the overlay)

When breeders are looking for the best single cross combinations, diallel designs have been by far the most used design in crops like maize. There are 4 types of diallel designs depending if reciprocate and self cross (omission of parents) are performed (full diallel with parents n^2; full diallel without parents n(n-1); half diallel with parents 1/2 * n(n+1); half diallel without parents 1/2 * n(n-1)). In this example we will show a full dialle design (reciprocate crosses are performed) and half diallel designs (only one of the directions is performed).

In the first data set we show a full diallel among 40 lines from 2 heterotic groups, 20 in each. Therefore 400 possible hybrids are possible. We have pehnotypic data for 100 of them across 4 locations. We use the data available to fit a model of the form:

$y = X\beta + Zu_1 + Zu_2 + Zu_S + \epsilon$

We estimate variance components for GCA_1, GCA_2 and SCA and use them to estimate heritability. Additionally BLUPs for GCA and SCA effects can be used to predict crosses.

```
data(cornHybrid)
hybrid2 <- cornHybrid$hybrid # extract cross data
head(hybrid2)
```

```
##   Location GCA1   GCA2         SCA Yield PlantHeight
## 1        1 A258 AS5707 A258:AS5707    NA          NA
## 2        1 A258     B2     A258:B2    NA          NA
## 3        1 A258    B99    A258:B99    NA          NA
## 4        1 A258   LH51   A258:LH51    NA          NA
## 5        1 A258   Mo44   A258:Mo44    NA          NA
## 6        1 A258  NC320  A258:NC320    NA          NA
```

```
modFD <- mmer2(Yield~Location,
               random=~GCA1+GCA2+SCA,
               rcov=~units,
               data=hybrid2,silent = TRUE, draw=FALSE)
(suma <- summary(modFD))
```

```
## ====================================================
##     Multivariate Linear Mixed Model fit by REML
## ******************  sommer 3.3  ******************
## ====================================================
##          logLik      AIC      BIC Method Converge
## Value -132.5889 273.1777 289.1436    MNR     TRUE
## ====================================================
## Variance-Covariance components:
##                    VarComp VarCompSE  Zratio
## GCA1.Yield-Yield     0.000       NaN  0.0000
## GCA2.Yield-Yield     7.302     18.88  0.3867
## SCA.Yield-Yield    187.652     41.62  4.5083
## units.Yield-Yield  221.142     18.15 12.1861
## ====================================================
## Fixed effects:
##
## $Yield
##                   Estimate Std. Error       t value
## (Intercept)  1.379351e+02   2.121492  6.501796e+01
## Location2   -1.669775e-13   2.103057 -7.939753e-14
## Location3    7.835337e+00   2.103057  3.725689e+00
## Location4   -9.097455e+00   2.103057 -4.325824e+00
##
## ====================================================
## Groups and observations:
##      Observ Groups
## GCA1    400     20
## GCA2    400     20
## SCA     400    400
## ====================================================
## Use the '$' sign to access results and parameters
```

```
Vgca <- sum(suma$var.comp.table[1:2,1])
Vsca <- suma$var.comp.table[3,1]
Ve <- suma$var.comp.table[4,1]
Va = 4*Vgca
Vd = 4*Vsca
Vg <- Va + Vd
(H2 <- Vg / (Vg + (Ve)) )
```

```
## [1] 0.7790693
```

```
(h2 <- Va / (Vg + (Ve)) )
```

```
## [1] 0.02917923
```

Don't worry too much about the small h2 value, the data was simulated to be mainly dominance variance, therefore the Va was simulated extremely small leading to such value of narrow sense h2.

In this second data set we show a small half diallel with 7 parents crossed in one direction. n(n-1)/2 crosses are possible 7(6)/2 = 21 unique crosses. Parents appear as males or females indistictly. Each with two replications in a CRD. For a half diallel design a single GCA variance component for both males and females can be estimated and an SCA as well ($\sigma^2_G CA$ and $\sigma^2_S CA$ respectively), and BLUPs for GCA and SCA of the parents can be extracted. We would show first how to use it with the `mmer2` function using the `overlay()` function and later we will show how to do it creating customized matrices using the `overlay` and `model.matrix` functions for the GCA and SCA matrices respectively. The specific model here is:

$y = X\beta + Zu_g + Zu_s + \epsilon$

```
data(HDdata)
head(HDdata)
```

```
##   rep geno male female     sugar
## 1   1   12    1      2 13.950509
## 2   2   12    1      2  9.756918
## 3   1   13    1      3 13.906355
## 4   2   13    1      3  9.119455
## 5   1   14    1      4  5.174483
## 6   2   14    1      4  8.452221
```

```
HDdata$geno <- as.factor(HDdata$geno)
HDdata$male <- as.factor(HDdata$male)
HDdata$female <- as.factor(HDdata$female)
# Fit the model
modHD <- mmer2(sugar~1,
               random=~overlay(male,female) + geno,
               rcov=~units,
               data=HDdata, silent = TRUE)
summary(modHD)
```

```
## ============================================================
##          Multivariate Linear Mixed Model fit by REML
## **********************  sommer 3.3  **********************
## ============================================================
##          logLik      AIC      BIC Method Converge
## Value -5.674408 13.34882 15.08649    MNR     TRUE
## ============================================================
## Variance-Covariance components:
##                                  VarComp VarCompSE Zratio
## overlay(male, female).sugar-sugar   5.509     3.579  1.539
## geno.sugar-sugar                    1.816     1.363  1.332
## units.sugar-sugar                   3.117     0.962  3.240
## ============================================================
## Fixed effects:
##
## $sugar
##           Estimate Std. Error  t value
## Intercept 10.33318   1.818944 5.680868
##
## ============================================================
## Groups and observations:
##                       Observ Groups
## overlay(male, female)     42      7
## geno                      42     21
## ============================================================
## Use the '$' sign to access results and parameters
```

```
suma <- summary(modHD)$var.comp.table
Vgca <- suma[1,1]
Vsca <- suma[2,1]
Ve <- suma[3,1]
Va = 4*Vgca
Vd = 4*Vsca
Vg <- Va + Vd
(H2 <- Vg / (Vg + (Ve/2)) ) # 2 technical reps
```

```
## [1] 0.9494886
```

```
(h2 <- Va / (Vg + (Ve/2)) )
```

```
## [1] 0.7140843
```

Notice how the `overlay()` argument makes the overlap of incidence matrices possible making sure that male and female are joint into a single random effect. The same can be done using the `mmer` argument by creating the incidence and covariance matrices in case you want to see what is doing `mmer2` in the background.

```
data(HDdata)
head(HDdata)
```

```
##   rep geno male female     sugar
## 1   1   12    1      2 13.950509
## 2   2   12    1      2  9.756918
## 3   1   13    1      3 13.906355
## 4   2   13    1      3  9.119455
## 5   1   14    1      4  5.174483
## 6   2   14    1      4  8.452221
```

```
#### GCA matrix for half diallel using male and female columns
#### use the 'overlay' function to create the half diallel matrix
Z1 <- overlay(HDdata$female, HDdata$male)
#### Obtain the SCA matrix
Z2 <- model.matrix(~as.factor(geno)-1, data=HDdata)
#### Define the response variable and run
y <- HDdata$sugar
ETA <- list(list(Z=Z1), list(Z=Z2)) # Zu component
modHD <- mmer(Y=y, Z=ETA, draw=FALSE, silent=TRUE)
summary(modHD)
```

```
## ====================================================
##     Multivariate Linear Mixed Model fit by REML
## ******************  sommer 3.3  ******************
## ====================================================
##          logLik      AIC      BIC Method Converge
## Value -5.674413 13.34883 15.08649    MNR     TRUE
## ====================================================
## Variance-Covariance components:
##             VarComp VarCompSE Zratio
## u1.T1-T1      5.507     3.577  1.539
## u2.T1-T1      1.820     1.367  1.331
## units.T1-T1   3.116     0.961  3.242
## ====================================================
## Fixed effects:
##
## $T1
##           Estimate Std. Error  t value
## Intercept 10.33318   1.818652 5.681781
##
## ====================================================
## Groups and observations:
##    Observ Groups
## u1     42      7
## u2     42     21
## ====================================================
## Use the '$' sign to access results and parameters
```

5) Genomic selection

In this section we will use wheat data from CIMMYT to show how is genomic selection performed. This is the case of prediction of specific individuals within a population. It basically uses a similar model of the form:

$y = X\beta + Zu + \epsilon$

and takes advantage of the variance covariance matrix for the genotype effect known as the additive relationship matrix (A) and calculated using the `A.mat` function to establish connections among all individuals and predict the BLUPs for individuals that were not measured. The prediction accuracy depends on several factors such as the heritability (h^2), training population used (TP), size of TP, etc.

```r
data(wheatLines);
X <- wheatLines$wheatGeno; X[1:5,1:4]; dim(X)
```

```
##      wPt.0538 wPt.8463 wPt.6348 wPt.9992
## [1,]       -1        1        1        1
## [2,]        1        1        1        1
## [3,]        1        1        1        1
## [4,]       -1        1        1        1
## [5,]       -1        1        1        1
```

```
## [1]  599 1279
```

```r
Y <- data.frame(wheatLines$wheatPheno); Y$id <- rownames(Y); head(Y);
```

```
##              X1          X2          X4         X5   id
## 775   1.6716295 -1.72746986 -1.89028479  0.0509159  775
## 2166 -0.2527028  0.40952243  0.30938553 -1.7387588 2166
## 2167  0.3418151 -0.64862633 -0.79955921 -1.0535691 2167
## 2465  0.7854395  0.09394919  0.57046773  0.5517574 2465
## 3881  0.9983176 -0.28248062  1.61868192 -0.1142848 3881
## 3889  2.3360969  0.62647587  0.07353311  0.7195856 3889
```

```r
rownames(X) <- rownames(Y)
# select environment 1
K <- A.mat(X) # additive relationship matrix
# GBLUP pedigree-based approach
set.seed(12345)
y.trn <- Y
vv <- sample(rownames(Y),round(dim(Y)[1]/5))
y.trn[vv,"X1"] <- NA
ans <- mmer2(X1~1,
             random=~g(id),
             rcov=~units,
             G=list(id=K),
             data=y.trn, silent = TRUE) # kinship based
cor(ans$u.hat$`g(id)`[vv,],Y[vv,"X1"])
```

```
## [1] 0.4885693
```

6) Single cross prediction

When doing prediction of single cross performance the phenotype can be dissected in three main components, the general combining abilities (GCA) and specific combining abilities (SCA). This can be expressed with the same model analyzed in the diallel experiment mentioned before:

$y = X\beta + Zu_1 + Zu_2 + Zu_S + \epsilon$

with:

u_1 ~ N(0, $K_1\sigma^2_u1$)

u_2 ~ N(0, $K_2\sigma^2_u2$)

u_s ~ N(0, $K_3\sigma^2_us$)

And we can specify the K matrices. The main difference between this model and the full and half diallel designs is the fact that this model will include variance covariance structures in each of the three random effects (GCA1, GCA2 and SCA) to be able to predict the crosses that have not ocurred yet. We will use the data published by Technow et al. (2015) to show how to do prediction of single crosses.

```
data(Technow_data)

A.flint <- Technow_data$AF # Additive relationship matrix Flint
A.dent <- Technow_data$AD # Additive relationship matrix Dent

pheno <- Technow_data$pheno # phenotypes for 1254 single cross hybrids
head(pheno);dim(pheno)
```

```
##    hybrid dent flint     GY    GM      hy
## 1 518.298  518   298  -8.04 -0.85 518:298
## 2 518.305  518   305 -11.10  1.70 518:305
## 3 518.306  518   306 -16.85  2.24 518:306
## 4 518.316  518   316   2.08 -1.33 518:316
## 5 518.323  518   323   5.65 -2.71 518:323
## 6 518.327  518   327 -16.95 -0.52 518:327
```

```
## [1] 1254    6
```

```
# CREATE A DATA FRAME WITH ALL POSSIBLE HYBRIDS
DD <- kronecker(A.dent,A.flint,make.dimnames=TRUE)
hybs <- data.frame(sca=rownames(DD),yield=NA,matter=NA,gcad=NA, gcaf=NA)
hybs$yield[match(pheno$hy, hybs$sca)] <- pheno$GY
hybs$matter[match(pheno$hy, hybs$sca)] <- pheno$GM
hybs$gcad <- as.factor(gsub(":.*","",hybs$sca))
hybs$gcaf <- as.factor(gsub(".*:","",hybs$sca))
head(hybs)
```

```
##       sca yield matter gcad gcaf
## 1 513:316 10.02  -2.05  513  316
## 2 513:323  6.97  -3.78  513  323
## 3 513:330    NA     NA  513  330
## 4 513:336    NA     NA  513  336
## 5 513:340    NA     NA  513  340
## 6 513:341    NA     NA  513  341
```

```
# RUN THE PREDICTION MODEL
y.trn <- hybs
vv1 <- which(!is.na(hybs$yield))
vv2 <- sample(vv1, 100)
y.trn[vv2,"yield"] <- NA
anss2 <- mmer2(yield~1,
               random=~g(gcad) + g(gcaf),
               rcov=~units,
               G=list(gcad=A.dent, gcaf=A.flint),
               method="NR", silent=TRUE, data=y.trn)
summary(anss2)
```

```
## ==================================================
##     Multivariate Linear Mixed Model fit by REML
## ******************  sommer 3.3  ******************
## ==================================================
##         logLik       AIC       BIC Method Converge
## Value 121.6303 -241.2605 -236.2095    MNR     TRUE
## ==================================================
## Variance-Covariance components:
##                      VarComp VarCompSE Zratio
## g(gcad).yield-yield   16.19    2.6079  6.206
## g(gcaf).yield-yield   11.27    2.1501  5.242
## units.yield-yield     17.65    0.8068 21.878
## ==================================================
## Fixed effects:
##
## $yield
##            Estimate Std. Error   t value
## Intercept 0.1245116  0.2035404 0.6117291
##
## ==================================================
## Groups and observations:
##         Observ Groups
## g(gcad)   1154    123
## g(gcaf)   1154     86
## ==================================================
## Use the '$' sign to access results and parameters
```

```
cor(anss2$fitted.y[vv2], hybs$yield[vv2])
```

```
## [1] 0.8797643
```

In the previous model we only used the GCA effects (GCA1 and GCA2) for practicity, altough it's been shown that the SCA effect doesn't actually help that much in increasing prediction accuracy and increase a lot the computation intensity required since the variance covariance matrix for SCA is the kronecker product of the variance covariance matrices for the GCA effects, resulting in a 10578x10578 matrix that increases in a very intensive manner the computation required.

A model without covariance structures would show that the SCA variance component is insignificant compared to the GCA effects. This is why including the third random effect doesn't increase the prediction accuracy.

7) Spatial modeling (using the 2-dimensional spline)

We will use the CPdata to show the use of 2-dimensional splines for accomodating spatial effects in field experiments. In early generation variety trials the availability of seed is low, which makes the use of unreplicated design a neccesity more than anything else. Experimental designs such as augmented designs and partially-replicated (p-rep) designs become every day more common this days.

In order to do a good job modeling the spatial trends happening in the field special covariance structures have been proposed to accomodate such spatial trends (i.e. autoregressive residuals; ar1). Unfortunately, some of these covariance structures make the modeling rather unstable. More recently other research groups have proposed the use of 2-dimensional splines to overcome such issues and have a more robust modeling of the spatial terms (Lee et al. 2013; Rodríguez-Álvarez et al. 2018).

In this example we assume an unreplicated population where row and range information is available which allows us to fit a 2 dimensional spline model.

```
data(CPdata)
head(CPpheno)
```

```
##        id Row Col Year      color  Yield FruitAver Firmness Rowf Colf
## P003 P003   3   1 2014 0.10075269 154.67     41.93  588.917    3    1
## P004 P004   4   1 2014 0.13891940 186.77     58.79  640.031    4    1
## P005 P005   5   1 2014 0.08681502  80.21     48.16  671.523    5    1
## P006 P006   6   1 2014 0.13408561 202.96     48.24  687.172    6    1
## P007 P007   7   1 2014 0.13519278 174.74     45.83  601.322    7    1
## P008 P008   8   1 2014 0.17406685 194.16     44.63  656.379    8    1
```

```
CPgeno[1:4,1:4]
```

```
##      scaffold_50439_2381 scaffold_39344_153 uneak_3436043 uneak_2632033
## P003                   0                  0             0             1
## P004                   0                  0             0             1
## P005                   0                 -1             0             1
## P006                  -1                 -1            -1             0
```

```
#### create the variance-covariance matrix
A <- A.mat(CPgeno) # additive relationship matrix
#### look at the data and fit the model
head(CPpheno)
```

```
##        id Row Col Year      color  Yield FruitAver Firmness Rowf Colf
## P003 P003   3   1 2014 0.10075269 154.67     41.93  588.917    3    1
## P004 P004   4   1 2014 0.13891940 186.77     58.79  640.031    4    1
## P005 P005   5   1 2014 0.08681502  80.21     48.16  671.523    5    1
## P006 P006   6   1 2014 0.13408561 202.96     48.24  687.172    6    1
## P007 P007   7   1 2014 0.13519278 174.74     45.83  601.322    7    1
## P008 P008   8   1 2014 0.17406685 194.16     44.63  656.379    8    1
```

```
mix1 <- mmer2(Yield~1,
              random=~g(id)
                      + Rowf + Colf
                      + spl2D(Row,Col),
              rcov=~units,
              G=list(id=A), silent=TRUE,
              data=CPpheno)
summary(mix1)
```

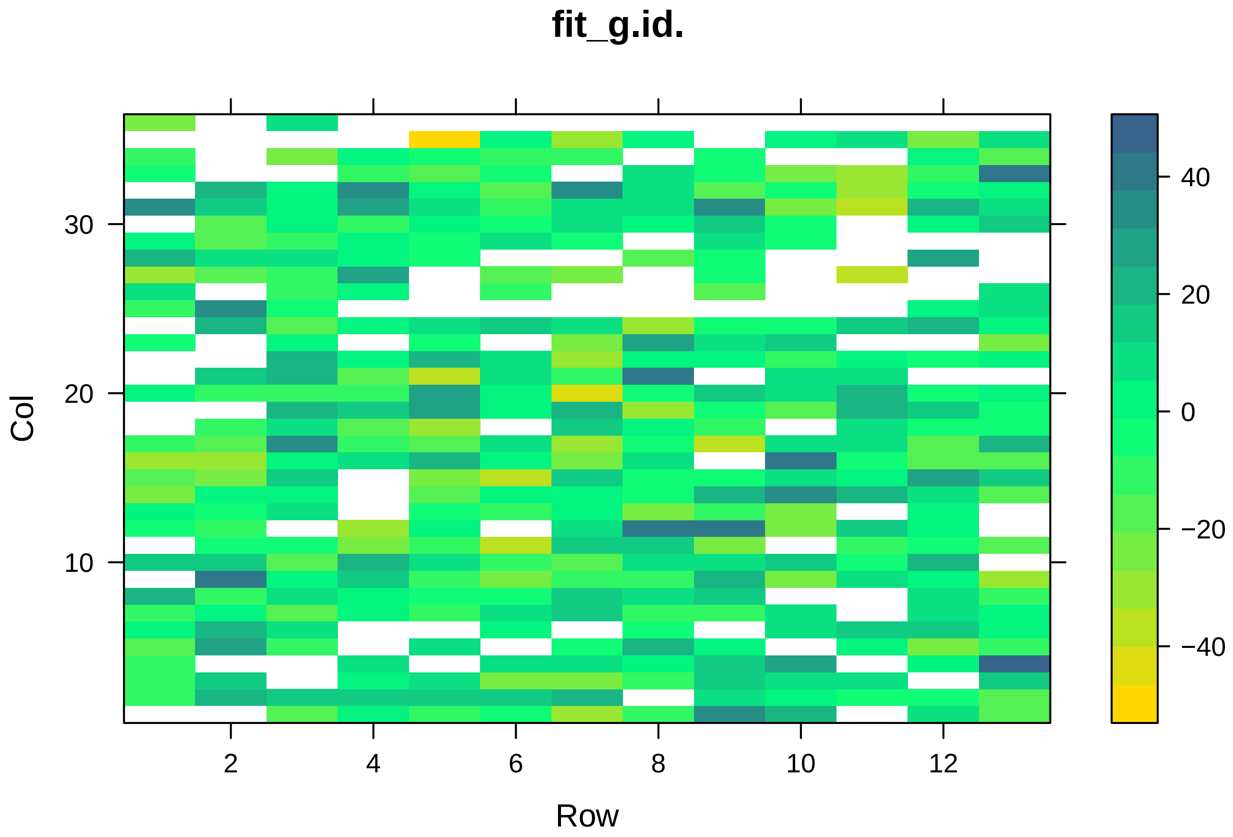
```
## =====================================================
##       Multivariate Linear Mixed Model fit by REML
## ********************  sommer 3.3  ********************
## =====================================================
##         logLik      AIC      BIC Method Converge
## Value -151.201 304.4021 308.2937    MNR     TRUE
## =====================================================
## Variance-Covariance components:
##                             VarComp VarCompSE Zratio
## g(id).Yield-Yield             782.9     318.9 2.4549
## Rowf.Yield-Yield              814.8     390.9 2.0846
## Colf.Yield-Yield              182.1     129.6 1.4056
## at_FIELD1_2Dspl.Yield-Yield   514.0     694.8 0.7397
## units.Yield-Yield            2922.8     294.2 9.9360
## =====================================================
## Fixed effects:
##
## $Yield
##           Estimate Std. Error  t value
## Intercept 132.1423   8.791225 15.03116
##
## =====================================================
## Groups and observations:
##                  Observ Groups
## g(id)              362    363
## Rowf               362     13
## Colf               362     36
## at_FIELD1_2Dspl    362    168
## =====================================================
## Use the '$' sign to access results and parameters
```

Notice that the job is done by the `spl2D()` function that takes the Row and Col information to fit a spatial kernel. When multiple fields are available the function has an additional argument call `at` which allows to fit a different spatial kernel in each field. For example if there was multiple fields the use of the random call would look like:

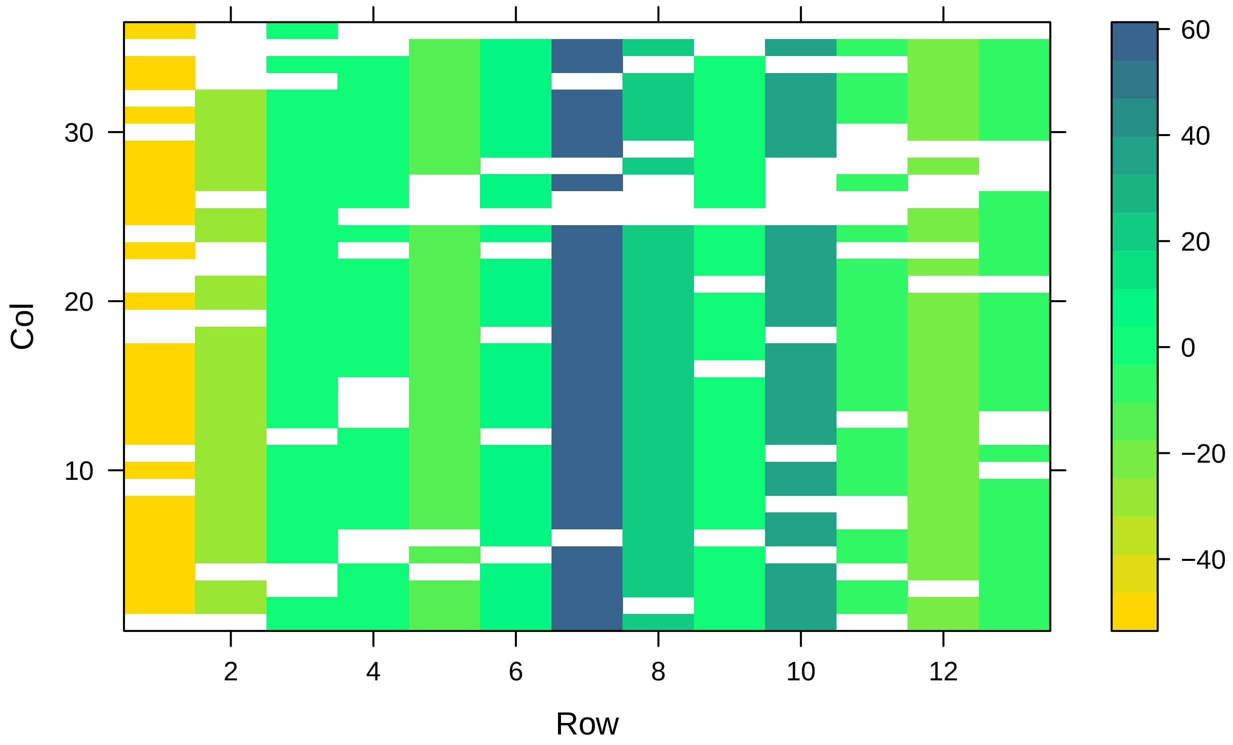
random=~ spl2D(Row,Col, at=FIELD)

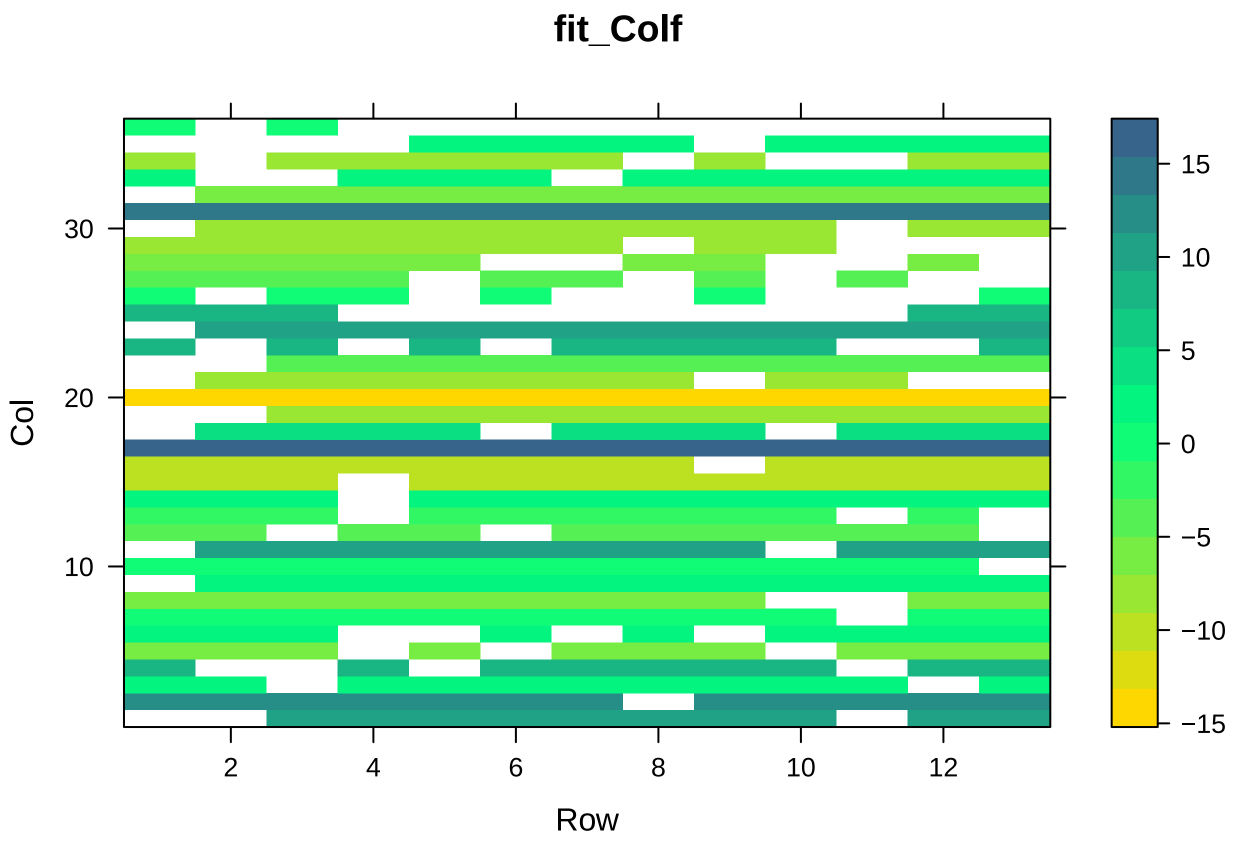
where FIELD would correspond to the name of the column of the dataset where the identifier for the different environments is.

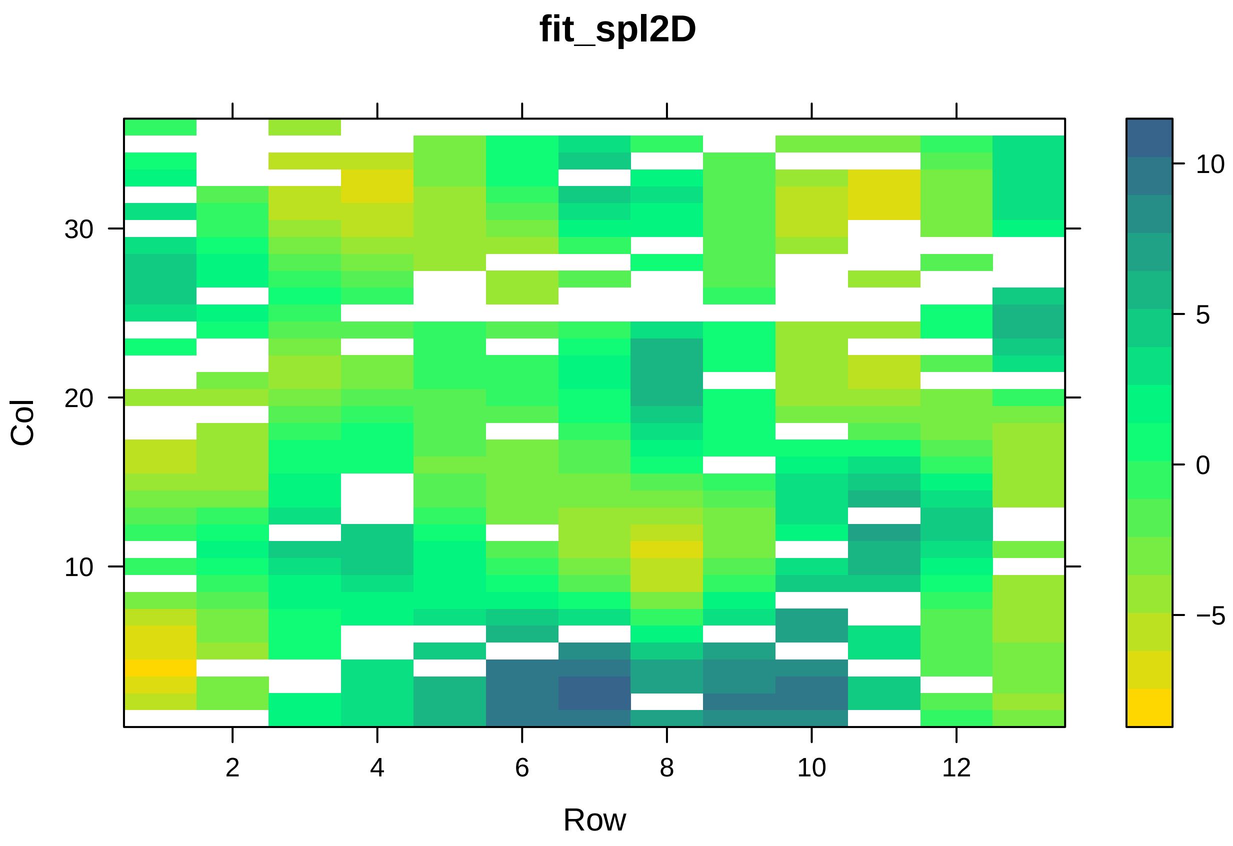
```
#### get the spatial plots
fittedvals <- spatPlots(mix1,row = "Row", range = "Col")
```

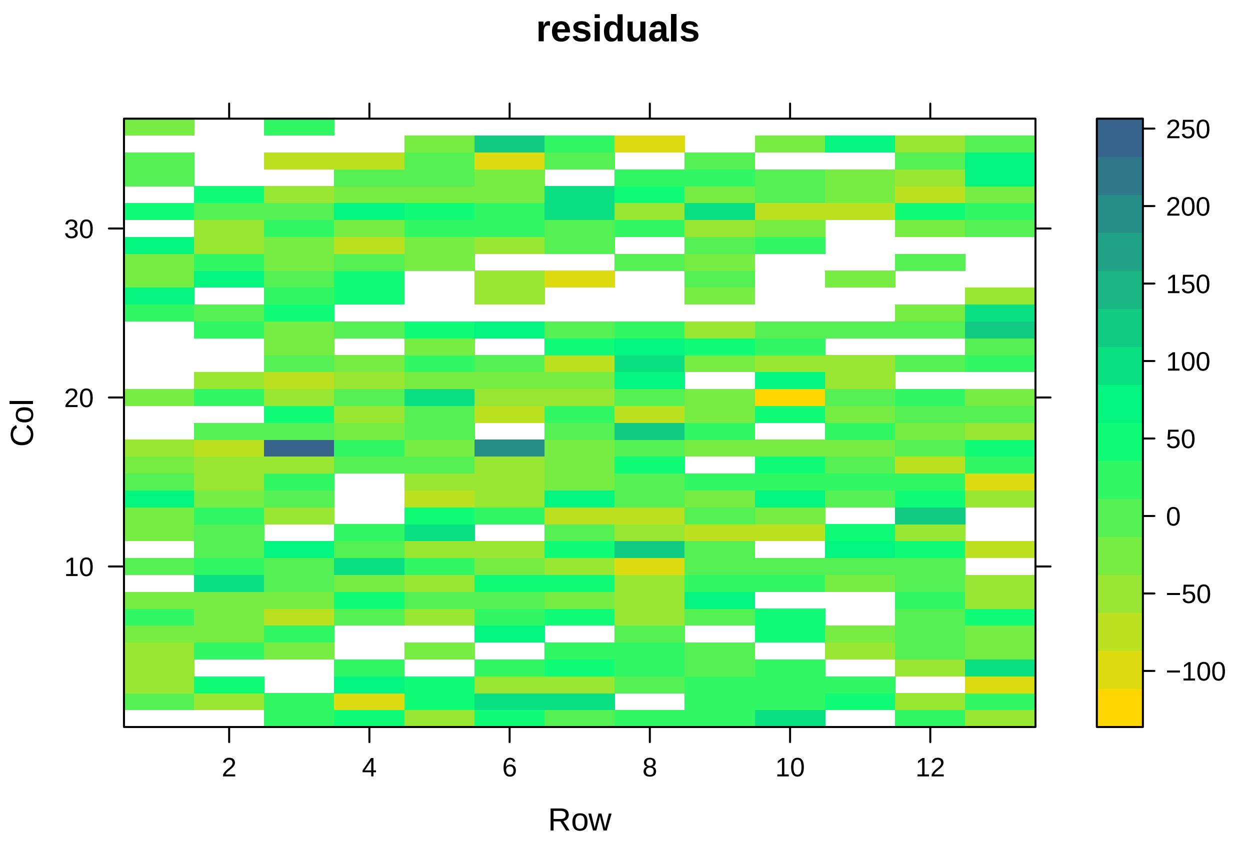

fit_g.id.
Col
30
20
10
Row
2
4
6
8
10
12
40
20
0
−20
−40

fit_Rowf




fit_Colf
Col
30
20
10
2
4
6
8
10
12
Row
15
10
5
0
−5
−10
−15


fit_spl2D
Col
10
20
30
Row
2
4
6
8
10
12
10
5
0
−5



8) Multivariate genetic models and genetic correlations

Sometimes is important to estimate genetic variance-covariance among traits, multi-reponse models are very useful for such task. Let see an example with 3 traits (color, Yield, and Firmness) and a single random effect (genotype; id) although multiple effects can be modeled as well. We need to use a variance covariance structure for the random effect to be able to obtain the genetic covariance among traits.

```
data(CPdata)
### look at the data
head(CPpheno);CPgeno[1:5,1:4]
```

```
##        id Row Col Year      color  Yield FruitAver Firmness Rowf Colf
## P003 P003   3   1 2014 0.10075269 154.67     41.93  588.917    3    1
## P004 P004   4   1 2014 0.13891940 186.77     58.79  640.031    4    1
## P005 P005   5   1 2014 0.08681502  80.21     48.16  671.523    5    1
## P006 P006   6   1 2014 0.13408561 202.96     48.24  687.172    6    1
## P007 P007   7   1 2014 0.13519278 174.74     45.83  601.322    7    1
## P008 P008   8   1 2014 0.17406685 194.16     44.63  656.379    8    1
```

```
##      scaffold_50439_2381 scaffold_39344_153 uneak_3436043 uneak_2632033
## P003                   0                  0             0             1
## P004                   0                  0             0             1
## P005                   0                 -1             0             1
## P006                  -1                 -1            -1             0
## P007                   0                  0             0             1
## fit a model including additive effects
A <- A.mat(CPgeno) # additive relationship matrix
####=================####
#### ADDITIVE MODEL ####
####=================####
ans.A <- mmer2(cbind(color,Yield)~1,
               random=~us(trait):g(id),
               rcov=~us(trait):units,
               G=list(id=A),
               data=CPpheno, silent = TRUE)
summary(ans.A)
```

```
## ==================================================
##     Multivariate Linear Mixed Model fit by REML
## ******************  sommer 3.3  ******************
## ==================================================
##          logLik      AIC      BIC Method Converge
## Value -286.6437 577.2875 586.4626    MNR     TRUE
## ==================================================
## Variance-Covariance components:
##                     VarComp VarCompSE  Zratio
## g(id).color-color 5.116e-03 1.037e-03  4.9331
## g(id).color-Yield 3.544e-01 4.303e-01  0.8236
## g(id).Yield-Yield 6.497e+02 3.235e+02  2.0082
## units.color-color 2.738e-03 2.994e-04  9.1477
## units.color-Yield 2.151e-01 2.268e-01  0.9484
## units.Yield-Yield 4.020e+03 3.429e+02 11.7223
## ==================================================
## Fixed effects:
##
## $color
##             Estimate  Std. Error   t value
## Intercept -0.7887719 0.002746636 -287.1775
##
## $Yield
##           Estimate Std. Error  t value
## Intercept 135.1766   3.327659 40.62214
##
## ==================================================
## Groups and observations:
##       Observ Groups
## g(id)    363    363
## ==================================================
## Use the '$' sign to access results and parameters
```

Now you can extract the BLUPs using the 'randef' function or simple accesing with the '$' sign and pick 'u.hat'. Also, genetic correlations and heritabilities can be calculated easily.

```
## genetic variance covariance
gvc <- ans.A$var.comp$`g(id)`
## extract variances (diagonals) and get standard deviations
sd.gvc <- as.matrix(sqrt(diag(gvc)))
## get possible products sd(Vgi) * sd(Vgi')
prod.sd <- sd.gvc %*% t(sd.gvc)
## genetic correlations cov(gi,gi')/[sd(Vgi) * sd(Vgi')]
```

```
(gen.cor <- gvc/prod.sd)
```

```
##           color     Yield
## color 1.0000000 0.1943762
## Yield 0.1943762 1.0000000
```

```
## heritabilities
(h2 <- diag(gvc) / diag(cov(CPpheno[,names(diag(gvc))], use = "complete.obs")))
```

```
##     color     Yield
## 0.7699871 0.1439069
```

Keep in mind that sommer uses direct inversion (DI) algorithm which can be very slow for large datasets. The package is focused in problems of the type p > n (more random effect levels than observations) and models with dense covariance structures. For example, for experiment with dense covariance structures with low-replication (i.e. 2000 records from 1000 individuals replicated twice with a covariance structure of 1000x1000) sommer will be faster than MME-based software. Also for genomic problems with large number of random effect levels, i.e. 300 individuals (n) with 100,000 genetic markers (p). For highly replicated trials with small covariance structures or n > p (i.e. 2000 records from 200 individuals replicated 10 times with covariance structure of 200x200) asreml or other MME-based algorithms will be much faster and we recommend you to opt for those software.

Literature

Covarrubias-Pazaran G. 2016. Genome assisted prediction of quantitative traits using the R package sommer. PLoS ONE 11(6):1-15.

Bernardo Rex. 2010. Breeding for quantitative traits in plants. Second edition. Stemma Press. 390 pp.

Gilmour et al. 1995. Average Information REML: An efficient algorithm for variance parameter estimation in linear mixed models. Biometrics 51(4):1440-1450.

Henderson C.R. 1975. Best Linear Unbiased Estimation and Prediction under a Selection Model. Biometrics vol. 31(2):423-447.

Kang et al. 2008. Efficient control of population structure in model organism association mapping. Genetics 178:1709-1723.

Lee, D.-J., Durban, M., and Eilers, P.H.C. (2013). Efficient two-dimensional smoothing with P-spline ANOVA mixed models and nested bases. Computational Statistics and Data Analysis, 61, 22 - 37.

Lee et al. 2015. MTG2: An efficient algorithm for multivariate linear mixed model analysis based on genomic information. Cold Spring Harbor. doi: http://dx.doi.org/10.1101/027201.

Maier et al. 2015. Joint analysis of psychiatric disorders increases accuracy of risk prediction for schizophrenia, bipolar disorder, and major depressive disorder. Am J Hum Genet; 96(2):283-294.

Rodríguez-Álvarez, María Xosé, et al. Correcting for spatial heterogeneity in plant breeding experiments with P-splines. Spatial Statistics 23 (2018): 52-71.

Searle. 1993. Applying the EM algorithm to calculating ML and REML estimates of variance components. Paper invited for the 1993 American Statistical Association Meeting, San Francisco.

Yu et al. 2006. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. Genetics 38:203-208.

Abdollahi Arpanahi R, Morota G, Valente BD, Kranis A, Rosa GJM, Gianola D. 2015. Assessment of bagging GBLUP for whole genome prediction of broiler chicken traits. Journal of Animal Breeding and Genetics 132:218-228.

Tunnicliffe W. 1989. On the use of marginal likelihood in time series model estimation. JRSS 51(1):15-27.