Special topics in Quantitative Genetics using sommer

Giovanny Covarrubias-Pazaran

2020-06-15

The sommer package was developed to provide R users a powerful and reliable multivariate mixed model solver. The package is focused in problems of the type p > n (more effects to estimate than observations) and its core algorithm is coded in C++ using the Armadillo library. This package allows the user to fit mixed models with the advantage of specifying the variance-covariance structure for the random effects, and specify heterogeneous variances, and obtain other parameters such as BLUPs, BLUEs, residuals, fitted values, variances for fixed and random effects, etc.

The purpose of this vignette is to show how to fit special models in quantitative genetics using the sommer package:

1) Partitioned model
2) UDU’ decomposition
3) Mating designs
4) Dominance variance

1) Partitioned model

The partitioned model was popularized by () to show that marker effects can be obtained fitting a GBLUP model to reduce the computational burden and then recover them by creating some special matrices MM’ for GBLUP and M’(M’M)- to recover marker effects. Here we show a very easy example using the DT_cpdata:

```r
library(sommer)
data("DT_cpdata")
DT <- DT_cpdata
M <- GT_cpdata

################
# MARKER MODEL
################
mix.marker <- mmer(color~1,
                   random=~Rowf+vs(M),
                   rcov=~units,data=DT,
                   verbose = FALSE)
```

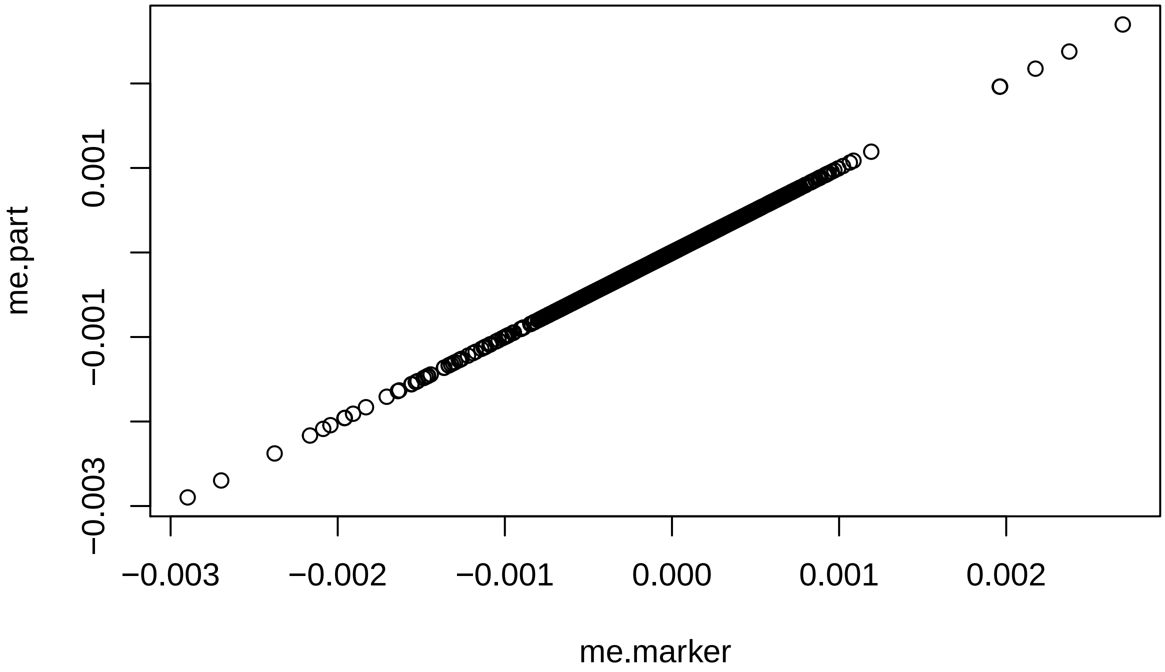
```
## Version out of date. Please update sommer to the newest version using:
## install.packages('sommer') in a new session
##  Use the 'date.warning' argument to disable the warning message.
```

```r
me.marker <- mix.marker$U$`u:M`$color

################
# PARTITIONED GBLUP MODEL
################
```

```r
MMT<-M%*%t(M) ## additive relationship matrix
MMTinv<-solve(MMT) ## inverse
MTMMTinv<-t(M)%*%MMTinv # M' %*% (M'M)-

mix.part <- mmer(color~1,
                 random=~Rowf+vs(id, Gu=MMT),
                 rcov=~units,data=DT,
                 verbose = FALSE)
```

```
## Version out of date. Please update sommer to the newest version using:
## install.packages('sommer') in a new session
##  Use the 'date.warning' argument to disable the warning message.
```

```r
#convert BLUPs to marker effects me=M'(M'M)- u
me.part<-MTMMTinv%*%matrix(mix.part$U$`u:id`$color,ncol=1)

# compare marker effects between both models
plot(me.marker,me.part)
```



As can be seen this two models are equivalent with the exception that the partitioned model is more computationally efficient.

2) UDU' decomposition

Lee and Van der Warf (2015) proposed a decomposition of the relationship matrix A=UDU' together with a transformation of the response and fixed effects Uy = Ux + UZ + e, to fit a model where the phenotypic variance matrix V is a diagonal because the relationship matrix is the diagonal matrix D from the decomposition that can be inverted easily and make multitrait models more feasible.

```r
data("DT_wheat")
rownames(GT_wheat) <- rownames(DT_wheat)
G <- A.mat(GT_wheat)
Y <- data.frame(DT_wheat)
```

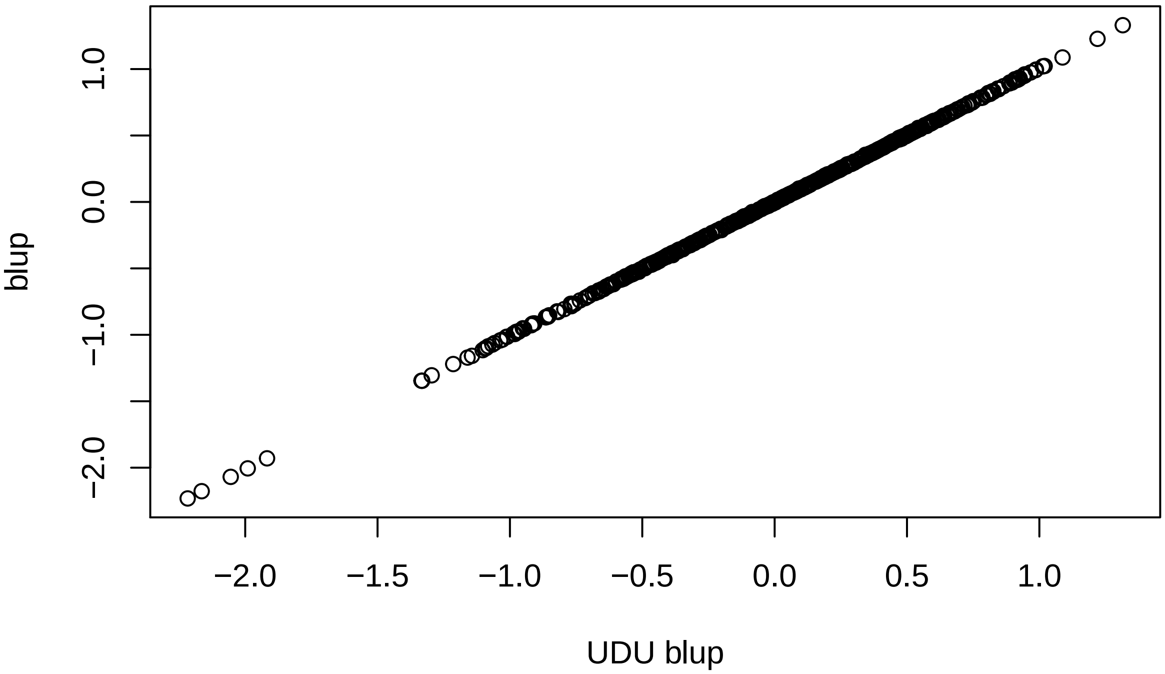
```r
# make the decomposition
UD<-eigen(G) # get the decomposition: G = UDU'
U<-UD$vectors
D<-diag(UD$values)# This will be our new 'relationship-matrix'
rownames(D) <- colnames(D) <- rownames(G)
X<-model.matrix(~1, data=Y) # here: only one fixed effect (intercept)
UX<-t(U)%*%X # premultiply X and y by U'
UY <- t(U) %*% as.matrix(Y) # multivariate

# dataset for decomposed model
DTd<-data.frame(id = rownames(G) ,UY, UX =UX[,1])
DTd$id<-as.character(DTd$id)

modeld <- mmer(cbind(X1,X2) ~ UX - 1,
              random = ~vs(id,Gu=D),
              rcov = ~vs(units),
              data=DTd, verbose = FALSE)
```

```
## Version out of date. Please update sommer to the newest version using:
## install.packages('sommer') in a new session
##  Use the 'date.warning' argument to disable the warning message.
```

```r
# dataset for normal model
DTn<-data.frame(id = rownames(G) , DT_wheat)
DTn$id<-as.character(DTn$id)

modeln <- mmer(cbind(X1,X2) ~ 1,
              random = ~vs(id,Gu=G),
              rcov = ~vs(units),
              data=DTn, verbose = FALSE)
```

```
## Version out of date. Please update sommer to the newest version using:
## install.packages('sommer') in a new session
##  Use the 'date.warning' argument to disable the warning message.
```

```r
## compare regular and transformed blups
plot(x=(solve(t(U)))%*%modeld$U$`u:id`$X2[colnames(D)],
     y=modeln$U$`u:id`$X2[colnames(D)], xlab="UDU blup",
     ylab="blup")
```



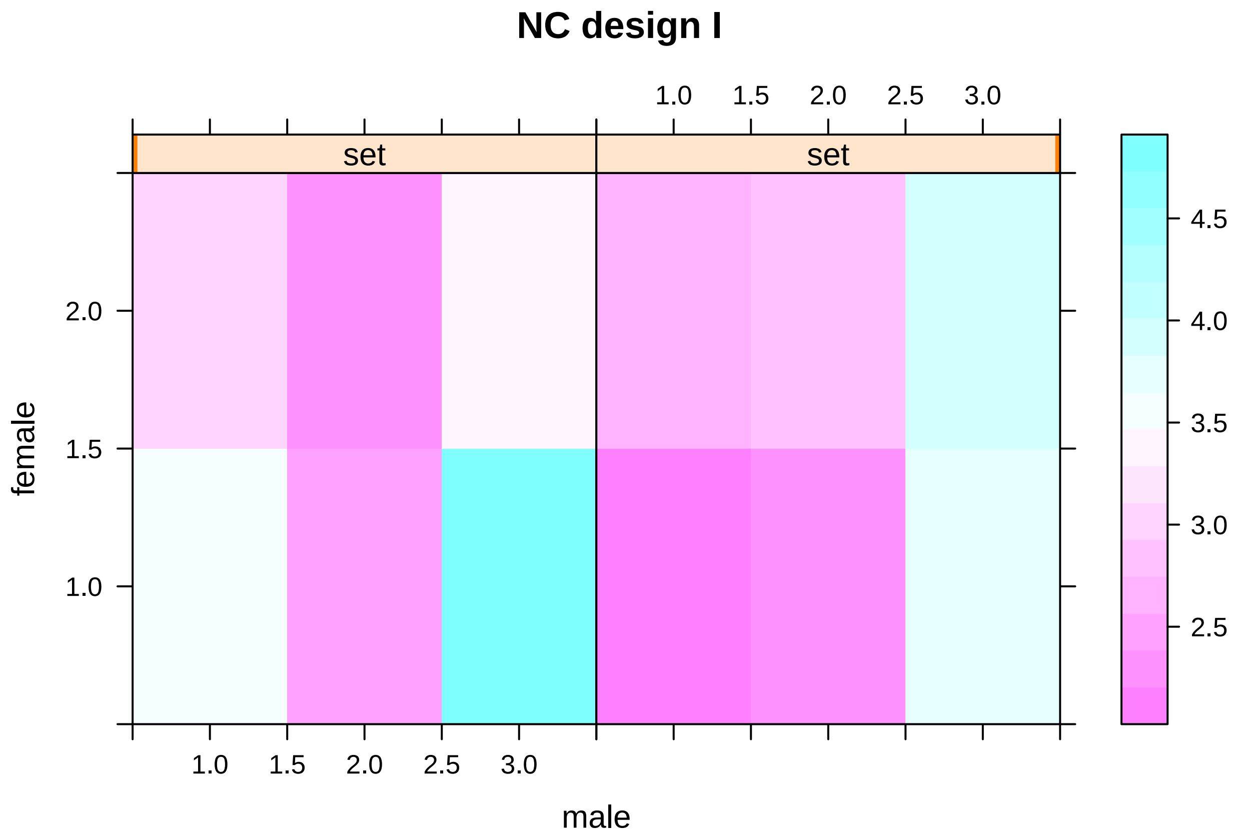
As can be seen the two models are equivalent. Despite that sommer doesnt take a great advantage of this trick because it was built for dense matrices using the Armadillo library other software may be better using this trick.

3) Mating designs

Estimating variance components has been a topic of interest for the breeding community for a long time. Here we show how to calculate additive and dominance variance using the North Carolina Design I (Nested design) and North Carolina Design II (Factorial design) using the classical Expected Mean Squares method and the REML methods from sommer and how these two are equivalent.

North Carolina Design I (Nested design)

```
data(DT_expdesigns)
DT <- DT_expdesigns$car1
DT <- aggregate(yield~set+male+female+rep, data=DT, FUN = mean)
DT$setf <- as.factor(DT$set)
DT$repf <- as.factor(DT$rep)
DT$malef <- as.factor(DT$male)
DT$femalef <- as.factor(DT$female)
levelplot(yield~male*female|set, data=DT, main="NC design I")
```



```
###############################
## Expected Mean Square method
###############################
mix1 <- lm(yield~ setf + setf:repf + femalef:malef:setf + malef:setf, data=DT)
MS <- anova(mix1); MS
```

```
## Analysis of Variance Table
##
## Response: yield
##                      Df Sum Sq Mean Sq F value   Pr(>F)
## setf                  1 0.1780 0.17796  1.6646 0.226012
## setf:repf             2 0.9965 0.49824  4.6605 0.037141 *
## setf:malef            4 7.3904 1.84759 17.2822 0.000173 ***
## setf:femalef:malef    6 1.6083 0.26806  2.5074 0.095575 .
## Residuals            10 1.0691 0.10691
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ms1 <- MS["malef:setf","Mean Sq"]
ms2 <- MS["femalef:malef:setf","Mean Sq"]
mse <- MS["Residuals","Mean Sq"]
nrep=2
nfem=2
Vfm <- (ms2-mse)/nrep
Vm <- (ms1-ms2)/(nrep*nfem)

## Calculate Va and Vd
Va=4*Vm # assuming no inbreeding (4/(1+F))
```

```
Vd=4*(Vfm-Vm) # assuming no inbreeding(4/(1+F)^2)
Vg=c(Va,Vd); names(Vg) <- c("Va","Vd"); Vg
```

```
## Va Vd
## NA NA
```

```
################################
## REML method
################################
mix2 <- mmer(yield~ setf + setf:repf,
            random=~femalef:malef:setf + malef:setf,
            data=DT, verbose = FALSE)
```

```
## Version out of date. Please update sommer to the newest version using:
## install.packages('sommer') in a new session
##  Use the 'date.warning' argument to disable the warning message.
```

```
vc <- summary(mix2)$varcomp; vc
```

```
##                                     VarComp  VarCompSE    Zratio Constraint
## femalef:malef:setf.yield-yield 0.08056338 0.08096526 0.9950364   Positive
## malef:setf.yield-yield         0.39480593 0.32832346 1.2024908   Positive
## units.yield-yield              0.10691762 0.04785610 2.2341480   Positive
```

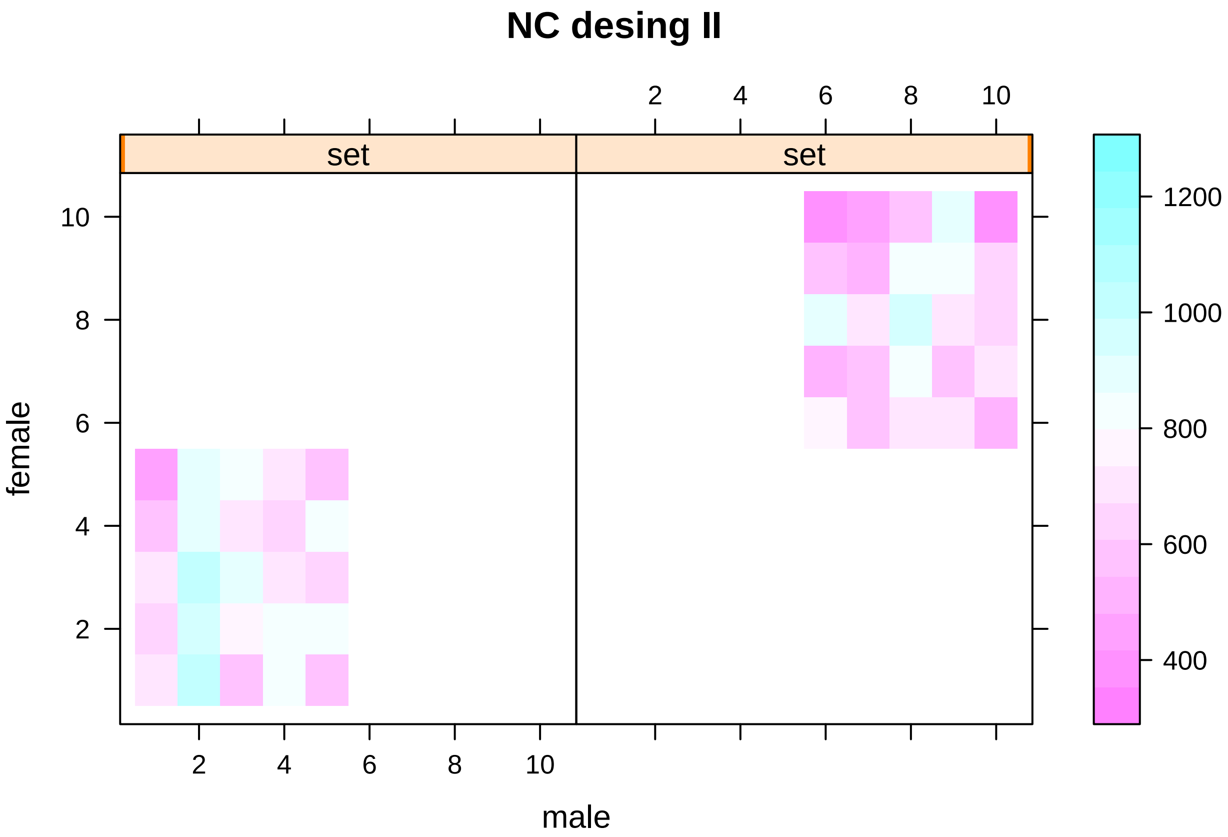
```
Vfm <- vc[1,"VarComp"]
Vm <- vc[2,"VarComp"]

## Calculate Va and Vd
Va=4*Vm # assuming no inbreeding (4/(1+F))
Vd=4*(Vfm-Vm) # assuming no inbreeding(4/(1+F)^2)
Vg=c(Va,Vd); names(Vg) <- c("Va","Vd"); Vg
```

```
##        Va        Vd
##  1.579224 -1.256970
```

As can be seen the REML method is easier than manipulating the MS and we arrive to the same results.

North Carolina Design II (Factorial design)

```
DT <- DT_expdesigns$car2
DT <- aggregate(yield~set+male+female+rep, data=DT, FUN = mean)
DT$setf <- as.factor(DT$set)
DT$repf <- as.factor(DT$rep)
DT$malef <- as.factor(DT$male)
DT$femalef <- as.factor(DT$female)
levelplot(yield~male*female|set, data=DT, main="NC desing II")
```



```
head(DT)
```

```
##   set male female rep   yield setf repf malef femalef
## 1   1    1      1   1  831.03    1    1     1       1
## 2   1    2      1   1 1046.55    1    1     2       1
## 3   1    3      1   1  853.33    1    1     3       1
## 4   1    4      1   1  940.00    1    1     4       1
## 5   1    5      1   1  802.00    1    1     5       1
## 6   1    1      2   1  625.93    1    1     1       2
```

```
N=with(DT,table(female, male, set))
nmale=length(which(N[1,,1] > 0))
nfemale=length(which(N[,1,1] > 0))
nrep=table(N[,,1])
nrep=as.numeric(names(nrep[which(names(nrep) !=0)]))

##############################
## Expected Mean Square method
##############################

mix1 <- lm(yield~ setf + setf:repf +
             femalef:malef:setf + malef:setf + femalef:setf, data=DT)
MS <- anova(mix1); MS
```

```
## Analysis of Variance Table
##
## Response: yield
##                     Df  Sum Sq Mean Sq F value    Pr(>F)
## setf                 1  847836  847836 45.6296 1.097e-09 ***
## setf:repf           4  144345   36086  1.9421  0.109652
## setf:malef          8  861053  107632  5.7926 5.032e-06 ***
## setf:femalef        8  527023   65878  3.5455  0.001227 **
## setf:femalef:malef 32  807267   25227  1.3577  0.129527
## Residuals          96 1783762   18581
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ms1 <- MS["setf:malef","Mean Sq"]
ms2 <- MS["setf:femalef","Mean Sq"]
ms3 <- MS["setf:femalef:malef","Mean Sq"]
mse <- MS["Residuals","Mean Sq"]
nrep=length(unique(DT$rep))
nfem=length(unique(DT$female))
nmal=length(unique(DT$male))
Vfm <- (ms3-mse)/nrep;
Vf <- (ms2-ms3)/(nrep*nmale);
Vm <- (ms1-ms3)/(nrep*nfemale);

Va=4*Vm; # assuming no inbreeding (4/(1+F))
Va=4*Vf; # assuming no inbreeding (4/(1+F))
Vd=4*(Vfm); # assuming no inbreeding(4/(1+F)^2)
Vg=c(Va,Vd); names(Vg) <- c("Va","Vd"); Vg
```

```
##        Va        Vd
## 10840.192  8861.659
```

```
##############################
## REML method
##############################

mix2 <- mmer(yield~ setf + setf:repf ,
            random=~femalef:malef:setf + malef:setf + femalef:setf,
            data=DT)
```

```
## Version out of date. Please update sommer to the newest version using:
## install.packages('sommer') in a new session
##  Use the 'date.warning' argument to disable the warning message.iteration    LogLik     wall    cpu(
##     1      -47.2423   21:10:27      0           0
##     2      -46.9375   21:10:27      0           0
##     3      -46.8452   21:10:27      0           0
##     4      -46.8361   21:10:27      0           0
##     5      -46.836   21:10:27      0           0
```

```
vc <- summary(mix2)$varcomp; vc
```

```
##                                  VarComp VarCompSE    Zratio Constraint
## femalef:malef:setf.yield-yield  2215.618  2284.794 0.9697231   Positive
## malef:setf.yield-yield          5493.338  3610.989 1.5212836   Positive
## femalef:setf.yield-yield        2710.176  2236.621 1.2117280   Positive
## units.yield-yield              18580.739  2681.742 6.9286068   Positive
```

```
Vfm <- vc[1,"VarComp"]
Vm <- vc[2,"VarComp"]
Vf <- vc[3,"VarComp"]

Va=4*Vm; # assuming no inbreeding (4/(1+F))
```

```
Va=4*Vf; # assuming no inbreeding (4/(1+F))
Vd=4*(Vfm); # assuming no inbreeding(4/(1+F)^2)
Vg=c(Va,Vd); names(Vg) <- c("Va","Vd"); Vg
```

```
##        Va        Vd
## 10840.704  8862.471
```

As can be seen the REML method is easier than manipulating the MS and we arrive to the same results.

4) Dominance variance

The estimation of non-additive variance has been proposed to be a challenge since the additive and dominance relationship matrices are not orthogonal. In recent literature it has been proposed that the best practice to fit the dominance component is to fit the additive component first and then fix the value of that variance c

```
data(DT_cpdata)
DT <- DT_cpdata
GT <- GT_cpdata
MP <- MP_cpdata
#### create the variance-covariance matrix
A <- A.mat(GT) # additive relationship matrix
#### look at the data and fit the model
mix1 <- mmer(Yield~1,
              random=~vs(id,Gu=A),
              rcov=~units,
              data=DT, verbose = FALSE)
```

```
## Version out of date. Please update sommer to the newest version using:
## install.packages('sommer') in a new session
##  Use the 'date.warning' argument to disable the warning message.
```

```
####=========================================####
#### adding dominance and forcing the other VC's
####=========================================####
DT$idd <- DT$id;
D <- D.mat(GT) # dominance relationship matrix
mm <- matrix(3,1,1) ## matrix to fix the var comp

mix2 <- mmer(Yield~1,
              random=~vs(id, Gu=A, Gt=mix1$sigma_scaled$`u:id`, Gtc=mm)
                      + vs(idd, Gu=D, Gtc=unsm(1)),
              rcov=~vs(units,Gt=mix1$sigma_scaled$units, Gtc=mm),
              data=DT, verbose = FALSE)
```

```
## Version out of date. Please update sommer to the newest version using:
## install.packages('sommer') in a new session
##  Use the 'date.warning' argument to disable the warning message.
```

```
# analyze variance components
summary(mix1)$varcomp
```

```
##                       VarComp VarCompSE    Zratio Constraint
## u:id.Yield-Yield     650.4145  325.5562  1.997856   Positive
## units.Yield-Yield   4031.0153  344.6051 11.697493   Positive
```

```
summary(mix2)$varcomp
```

```
##                          VarComp VarCompSE     Zratio Constraint
## u:id.Yield-Yield        650.4145  504.0361  1.2904126      Fixed
## u:idd.Yield-Yield       220.6311  410.7679  0.5371186   Positive
## u:units.Yield-Yield    4031.0153  360.7322 11.1745357      Fixed
```

Literature

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

Covarrubias-Pazaran G. 2018. Software update: Moving the R package sommer to multivariate mixed models for genome-assisted prediction. doi: https://doi.org/10.1101/354639

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.

Rodriguez-Alvarez, Maria Xose, 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.