Fitting genotype by environment models in lme4breeding

Giovanny Covarrubias-Pazaran

2024-05-16

The purpose of this vignette is to show how to fit different genotype by environment (GxE) models using the lme4breeding package:

1) Multienvironment model: Main effect model
2) Multienvironment model: Diagonal model (DG)
3) Multienvironment model: Compund symmetry model (CS) 3.2) Multienvironment model: Compund symmetry model + Diagonal (CS+DG)
4) Multienvironment model: Unstructured model (US)
5) Multienvironment model: Random regression model (RR) 5.2) Multienvironment model: Finlay-Wilkinson regression
6) Multienvironment model: Factor analytic (reduced rank) model (FA)
7) Two stage analysis

When the breeder decides to run a trial and apply selection in a single environment (whether because the amount of seed is a limitation or there's no availability for a location) the breeder takes the risk of selecting material for a target population of environments (TPEs) using an environment that is not representative of the larger TPE. Therefore, many breeding programs try to base their selection decision on multi-environment trial (MET) data. Models could be adjusted by adding additional information like spatial information, experimental design information, etc. In this tutorial we will focus mainly on the covariance structures for GxE and the incorporation of relationship matrices for the genotype effect.

1) MET: main effect model

A multi-environment model is the one that is fitted when the breeding program can afford more than one location. The main effect model assumes that GxE doesn't exist and that the main genotype effect plus the fixed effect for environment is enough to predict the genotype effect in all locations of interest.

```
library(lme4breeding)
data(DT_example)
DT <- DT_example
A <- A_example

ansMain <- lmebreed(Yield ~ Env + (1|Name),
                        relmat = list(Name = A ),
                        data=DT)
vc <- VarCorr(ansMain); print(vc,comp=c("Variance"))
```

```
##  Groups   Name        Variance
##  Name     (Intercept) 4.8559
##  Residual             8.1086
```

In this model, the only term to be estimated is the one for the germplasm (here called `Name`). For the sake of example we have added a relationship matrix among the levels of the random effect `Name`. This is just a

diagonal matrix with as many rows and columns as levels present in the random effect `Name`, but any other non-diagonal relationship matrix could be used.

2) MET: diagonal model (DG)

A multi-environment model is the one that is fitted when the breeding program can afford more than one location. The diagonal model assumes that GxE exists and that the genotype variation is expressed differently at each location, therefore fitting a variance component for the genotype effect at each location. The main drawback is that this model assumes no covariance among locations, as if genotypes were independent (despite the fact that is the same genotypes). The fixed effect for environment plus the location-specific BLUP is used to predict the genotype effect in each locations of interest.

```
Z <- with(DT, dsc(Env))$Z
diagFormula <- paste0( "Yield ~ Env + (0+", paste(colnames(Z), collapse = "+"), "|| Name)")
for(i in 1:ncol(Z)){DT[,colnames(Z)[i]] <- Z[,i]}
print(as.formula(diagFormula))
```

```
## Yield ~ Env + (0 + CA.2011 + CA.2012 + CA.2013 || Name)
```

```
ansDG <- lmebreed(as.formula(diagFormula),
                      relmat = list(Name = A ),
                      data=DT)
vc <- VarCorr(ansDG); print(vc,comp=c("Variance"))
```

```
##  Groups   Name    Variance
##  Name     CA.2011 17.4934
##  Name.1   CA.2012  5.3376
##  Name.2   CA.2013  7.8838
##  Residual          4.3806
```

```
ve <- attr(vc, "sc")^2; ve
```

```
## [1] 4.380621
```

3) MET: compund symmetry model (CS)

A multi-environment model is the one that is fitted when the breeding program can afford more than one location. The compound symmetry model assumes that GxE exists and that a main genotype variance-covariance component is expressed across all location. In addition, it assumes that a main genotype-by-environment variance is expressed across all locations. The main drawback is that the model assumes the same variance and covariance among locations. The fixed effect for environment plus the main effect for BLUP plus genotype-by-environment effect is used to predict the genotype effect in each location of interest.

```
DT$EnvName <- paste(DT$Env, DT$Name, sep = ":")
E <- Matrix::Diagonal(length(unique(DT$Env)));
colnames(E) <- rownames(E) <- unique(DT$Env);E
```

```
## 3 x 3 diagonal matrix of class "ddiMatrix"
##         CA.2013 CA.2011 CA.2012
## CA.2013       1       .       .
## CA.2011       .       1       .
## CA.2012       .       .       1
```

```
EA <- Matrix::kronecker(E,A, make.dimnames = TRUE)
ansCS <- lmebreed(Yield ~ Env + (1|Name) + (1|EnvName),
                    relmat = list(Name = A, EnvName=  EA ),
```

```
                    data=DT)
vc <- VarCorr(ansCS); print(vc,comp=c("Variance"))
```

```
##  Groups   Name        Variance
##  EnvName  (Intercept) 5.1732
##  Name     (Intercept) 3.6819
##  Residual             4.3662
```

```
ve <- attr(vc, "sc")^2; ve
```

```
## [1] 4.366211
```

3.2) MET: compund symmetry model + diagonal (CS+DG)

A multi-environment model is the one that is fitted when the breeding program can afford more than one location. The compound symmetry model assumes that GxE exists and that a main genotype variance-covariance component is expressed across all location. In addition, it assumes that a main genotype-by-environment variance is expressed across all locations. The main drawback is that the model assumes the same variance and covariance among locations. The fixed effect for environment plus the main effect for BLUP plus genotype-by-environment effect is used to predict the genotype effect in each location of interest.

```
Z <- with(DT, dsc(Env))$Z
csdiagFormula <- paste0( "Yield ~ Env + (", paste(colnames(Z), collapse = "+"), "|| Name)")
for(i in 1:ncol(Z)){DT[,colnames(Z)[i]] <- Z[,i]}
print(as.formula(csdiagFormula))
```

```
## Yield ~ Env + (CA.2011 + CA.2012 + CA.2013 || Name)
```

```
ansCSDG <- lmebreed(as.formula(csdiagFormula),
                    relmat = list(Name = A ),
                    data=DT)
vc <- VarCorr(ansCSDG); print(vc,comp=c("Variance"))
```

```
##  Groups   Name        Variance
##  Name     (Intercept)  2.9638
##  Name.1   CA.2011     10.4259
##  Name.2   CA.2012      2.6589
##  Name.3   CA.2013      5.7021
##  Residual              4.3976
```

```
ve <- attr(vc, "sc")^2; ve
```

```
## [1] 4.397563
```

4) MET: unstructured model (US)

A multi-environment model is the one that is fitted when the breeding program can afford more than one location. The unstructured model is the most flexible model assuming that GxE exists and that an environment-specific variance exists in addition to as many covariances for each environment-to-environment combinations. The main drawback is that is difficult to make this models converge because of the large number of variance components, the fact that some of these variance or covariance components are zero, and the difficulty in choosing good starting values. The fixed effect for environment plus the environment specific BLUP (adjusted by covariances) is used to predict the genotype effect in each location of interest.

```
Z <- with(DT, dsc(Env))$Z
usFormula <- paste0( "Yield ~ Env + (0+", paste(colnames(Z), collapse = "+"), "| Name)")
```

```
for(i in 1:ncol(Z)){DT[,colnames(Z)[i]] <- Z[,i]}
print(as.formula(usFormula))
```

```
## Yield ~ Env + (0 + CA.2011 + CA.2012 + CA.2013 | Name)
```

```
ansDG <- lmebreed(as.formula(usFormula),
                  relmat = list(Name = A ),
                  data=DT)
vc <- VarCorr(ansDG); print(vc,comp=c("Variance"))
```

```
##  Groups   Name    Variance Cov
##  Name     CA.2011 15.9937
##           CA.2012  5.2743   6.173
##           CA.2013  7.6897   6.366  0.375
##  Residual          4.3862
```

```
ve <- attr(vc, "sc")^2; ve
```

```
## [1] 4.386173
```

5) MET: random regression model

A multi-environment model is the one that is fitted when the breeding program can afford more than one location. The random regression model assumes that the environment can be seen as a continuous variable and therefore a variance component for the intercept and a variance component for the slope can be fitted. The number of variance components will depend on the order of the Legendre polynomial fitted.

```
# library(orthopolynom)
# DT$EnvN <- as.numeric(as.factor(DT$Env))
#
# Z <- with(DT, dsc(leg(EnvN,1)) )$Z
# rrFormula <- paste0( "Yield ~ Env + (0+", paste(colnames(Z), collapse = "+"), "| Name)")
# for(i in 1:ncol(Z)){DT[,colnames(Z)[i]] <- Z[,i]}
# ansRR <- lmebreed(as.formula(rrFormula),
#                   relmat = list(Name = A ),
#                   data=DT)
# vc <- VarCorr(ansRR); print(vc,comp=c("Variance"))
# ve <- attr(vc, "sc")^2; ve
```

In addition, we an fit this without covariance:

```
# library(orthopolynom)
# DT$EnvN <- as.numeric(as.factor(DT$Env))
#  Z <- with(DT, dsc(leg(EnvN,1)) )$Z
# rrFormula <- paste0( "Yield ~ Env + (0+", paste(colnames(Z), collapse = "+"), "|| Name)")
# for(i in 1:ncol(Z)){DT[,colnames(Z)[i]] <- Z[,i]}
# ansRR <- lmebreed(as.formula(rrFormula),
#                   relmat = list(Name = A ),
#                   data=DT)
# vc <- VarCorr(ansRR); print(vc,comp=c("Variance"))
# ve <- attr(vc, "sc")^2; ve
```

5.2) Finlay-Wilkinson regression

```
data(DT_h2)
DT <- DT_h2
## build the environmental index
ei <- aggregate(y~Env, data=DT,FUN=mean)
colnames(ei)[2] <- "envIndex"
ei$envIndex <- ei$envIndex - mean(ei$envIndex,na.rm=TRUE) # center the envIndex to have clean VCs
ei <- ei[with(ei, order(envIndex)), ]
## add the environmental index to the original dataset
DT2 <- merge(DT,ei, by="Env")
DT2 <- DT2[with(DT2, order(Name)), ]

ansFW <- lmebreed(y~ Env + (envIndex || Name), data=DT2)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model failed to conve
```

```
vc <- VarCorr(ansFW); print(vc,comp=c("Variance"))
```

```
##  Groups   Name        Variance
##  Name     (Intercept) 2.632643
##  Name.1   envIndex    0.041216
##  Residual             7.046535
```

```
ve <- attr(vc, "sc")^2; ve
```

```
## [1] 7.046535
```

Alternatively, you can also add the covariance between both the main effect and the sensitivity

```
ansFW2 <- lmebreed(y~ Env + (envIndex | Name), data=DT2)
vc <- VarCorr(ansFW2); print(vc,comp=c("Variance"))
```

```
##  Groups   Name        Variance Cov
##  Name     (Intercept) 2.75579
##           envIndex    0.04279  0.343
##  Residual             7.03681
```

```
ve <- attr(vc, "sc")^2; ve
```

```
## [1] 7.036805
```

6) Factor analytic (reduced rank) model

When the number of environments where genotypes are evaluated is big and we want to consider the genetic covariance between environments and location-specific variance components we cannot fit an unstructured covariance in the model since the number of parameters is too big and the matrix can become non-full rank leading to singularities. In those cases is suggested a dimensionality reduction technique. Among those the factor analytic structures proposed by many research groups (Piepho, Smith, Cullis, Thompson, Meyer, etc.) are the way to go. lme4breeding has a reduced-rank factor analytic implementation available through the rrc() function. Here we show an example of how to fit the model:

```
data(DT_h2)
DT <- DT_h2
DT=DT[with(DT, order(Env)), ]

# fit diagonal model first to produce H matrix
```

```
Z <- with(DT, dsc(Env))$Z
diagFormula <- paste0( "y ~ Env + (0+", paste(colnames(Z), collapse = "+"), "|| Name)")
for(i in 1:ncol(Z)){DT[,colnames(Z)[i]] <- Z[,i]}
print(as.formula(diagFormula))
```

```
## y ~ Env + (0 + CA.2011 + CA.2012 + CA.2013 + FL.2011 + FL.2012 +
##     FL.2013 + MI.2011 + MI.2012 + MI.2013 + MO.2011 + MO.2012 +
##     MO.2013 + NY.2011 + NY.2012 + NY.2013 || Name)
```

```
ans1a <- lmebreed(as.formula(diagFormula),
                  relmat = list(Name = A ),
                  data=DT)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
vc <- VarCorr(ans1a); print(vc,comp=c("Variance"))
```

```
##  Groups   Name    Variance
##  Name     CA.2011 1.7594e+01
##  Name.1   CA.2012 5.4589e+00
##  Name.2   CA.2013 7.9867e+00
##  Name.3   FL.2011 1.4378e+00
##  Name.4   FL.2012 5.5710e-09
##  Name.5   FL.2013 1.7613e-01
##  Name.6   MI.2011 8.3056e+00
##  Name.7   MI.2012 5.0141e+00
##  Name.8   MI.2013 1.9452e+01
##  Name.9   MO.2011 2.5298e-01
##  Name.10  MO.2012 1.5656e+01
##  Name.11  MO.2013 1.7901e+00
##  Name.12  NY.2011 4.3956e+00
##  Name.13  NY.2012 2.0397e+00
##  Name.14  NY.2013 8.5124e+00
##  Residual         4.1736e+00
```

```
H0 <- ranef(ans1a)$Name # GxE table

# reduced rank model
Z <- with(DT,  dsc(rrc(Env, H = H0, nPC = 3)) )$Z
Zd <- with(DT, dsc(Env))$Z
faFormula <- paste0( "y ~ Env + (0+", paste(colnames(Z), collapse = "+"), "| Name) + (0+",paste(colname
for(i in 1:ncol(Z)){DT[,colnames(Z)[i]] <- Z[,i]}
print(as.formula(faFormula))
```

```
## y ~ Env + (0 + PC1 + PC2 + PC3 | Name) + (0 + CA.2011 + CA.2012 +
##     CA.2013 + FL.2011 + FL.2012 + FL.2013 + MI.2011 + MI.2012 +
##     MI.2013 + MO.2011 + MO.2012 + MO.2013 + NY.2011 + NY.2012 +
##     NY.2013 || Name)
```

```
ansFA <- lmebreed(as.formula(faFormula),
                  relmat = list(Name = A ),
                  data=DT)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
vc <- VarCorr(ansFA); print(vc,comp=c("Variance"))
```

```
##  Groups   Name    Variance   Cov
```

```
##  Name     PC1     4.1016e+00
##           PC2     4.1789e+00  1.352
##           PC3     5.8003e+00  2.418 -1.411
##  Name.1   CA.2011 7.9845e+00
##  Name.2   CA.2012 0.0000e+00
##  Name.3   CA.2013 7.3062e-01
##  Name.4   FL.2011 1.1130e+00
##  Name.5   FL.2012 7.4207e-10
##  Name.6   FL.2013 2.3911e-09
##  Name.7   MI.2011 4.5986e+00
##  Name.8   MI.2012 2.7527e+00
##  Name.9   MI.2013 1.4452e+01
##  Name.10  MO.2011 0.0000e+00
##  Name.11  MO.2012 1.2246e+01
##  Name.12  MO.2013 1.9037e-01
##  Name.13  NY.2011 3.8832e+00
##  Name.14  NY.2012 2.0276e-01
##  Name.15  NY.2013 7.7721e+00
##  Residual         4.0236e+00
```

```
ve <- attr(vc, "sc")^2; ve
```

```
## [1] 4.0236
```

```
loadings=with(DT, rrc(Env, nPC = 3, H = H0, returnGamma = T) )$Gamma
Gint <- loadings %*% vc$Name %*% t(loadings)
Gspec <- diag( unlist(lapply(vc[2:16], function(x){x[[1]]})) )
G <- Gint + Gspec
# lattice::levelplot(cov2cor(G))
# colfunc <- colorRampPalette(c("steelblue4","springgreen","yellow"))
# hv <- heatmap(cov2cor(G), col = colfunc(100), symm = TRUE)

u <- ranef(ansFA)$Name
uInter <- as.matrix(u[,1:3]) %*% t(as.matrix(loadings))
uSpec <- as.matrix(u[,-c(1:3)])
u <- uSpec + uInter
```

As can be seen genotype BLUPs for all environments can be recovered by multiplying the loadings (Gamma) by the factor scores. This is a parsomonious way to model an unstructured covariance.

7) Two stage analysis

It is common then to fit a first model that accounts for the variation of random design elements, e.g., locations, years, blocks, and fixed genotype effects to obtain the estimated marginal means (EMMs) or best linear unbiased estimators (BLUEs) as adjusted entry means. These adjusted entry means are then used as the phenotype or response variable in GWAS and genomic prediction studies.

```
##########
## stage 1
##########
data(DT_h2)
DT <- DT_h2
head(DT)
```

```
##                 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
```

```
envs <- unique(DT$Env)
vals <- list()
for(i in 1:length(envs)){
  ans1 <- lmebreed(y~Name + (1|Block), data= droplevels(DT[which(DT$Env == envs[i]),]) )
  b <- fixef(ans1)
  b[2:length(b)] <- b[2:length(b)] + b[1]
  ids <- colnames(model.matrix(~Name-1, data=droplevels(DT[which(DT$Env == envs[i]),]) ))
  ids <- gsub("Name","",ids)
  vals[[i]] <- data.frame(Estimate=b , stdError= diag( vcov(ans1)), Effect= ids, Env= envs[i])
}
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
DT2 <- do.call(rbind, vals)

##########
## stage 2
##########
DT2$w <- 1/DT2$stdError
ans2 <- lmebreed(Estimate~Env + (1|Effect) + (1|Env:Effect), weights = w,data=DT2,
                 control = lmerControl(
                   # optimizer="bobyqa",
                   check.nobs.vs.nlev = "ignore",
                   check.nobs.vs.rankZ = "ignore",
                   check.nobs.vs.nRE="ignore"
                 )
                 )
vc <- VarCorr(ans2); print(vc,comp=c("Variance"))
```

```
##  Groups     Name        Variance
##  Env:Effect (Intercept) 2.91649
##  Effect     (Intercept) 1.99225
##  Residual               0.72139
```

```
ve <- attr(vc, "sc")^2; ve
```

```
## [1] 0.7213944
```

Literature

Giovanny Covarrubias-Pazaran (2024). lme4breeding: enabling genetic evaluation in the age of genomic data. To be submitted to Bioinformatics.

Bates Douglas, Maechler Martin, Bolker Ben, Walker Steve. 2015. Fitting Linear Mixed-Effects Models Using

lme4. Journal of Statistical Software, 67(1), 1-48.

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.

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