



WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Mixed Effects Model: Applications in R

FA21 ANSCI 610



Mixed Model Equations

Standard approach in modern animal breeding programs. Robust methodology for dealing with complex structure arising in field collected data.



Mixed Effects Models can accommodate different situations; *e.g.* unbalanced data, sex-limited traits, selection bias



The underlying genetic merit of animals without observation can be predicted based on their relatives performance



Efficient estimation of fixed (BLUE) and random (BLUP) effects, and (co)variance components that will guide the breeding design.



The animal model can be described in matrix notation as:

$$y = X\beta + Za + e$$

y ($n \times 1$) is a vector of observations (phenotypic scores)

X ($n \times p$) is an incidence matrix for the fixed effects

β ($p \times 1$) is a vector of fixed effects (e.g. herd-year-season effects)

Z ($n \times q$) is an incidence matrix linking the animals to their observed responses

$a \sim N(0, A\sigma_a^2)$ is the ($q \times 1$) vector of breeding values (relative to all individuals with record or in the pedigree file, such that q is in general bigger than n)

$e \sim N(0, I\sigma_e^2)$ is the $n \times 1$ vector of residual effects



Examples of fixed effects:

Farm
Herd
Season
Year
Nutritional Management
Climate variables
Age
Sex
Barn/Pen



Quick review on mixed model equations derivation

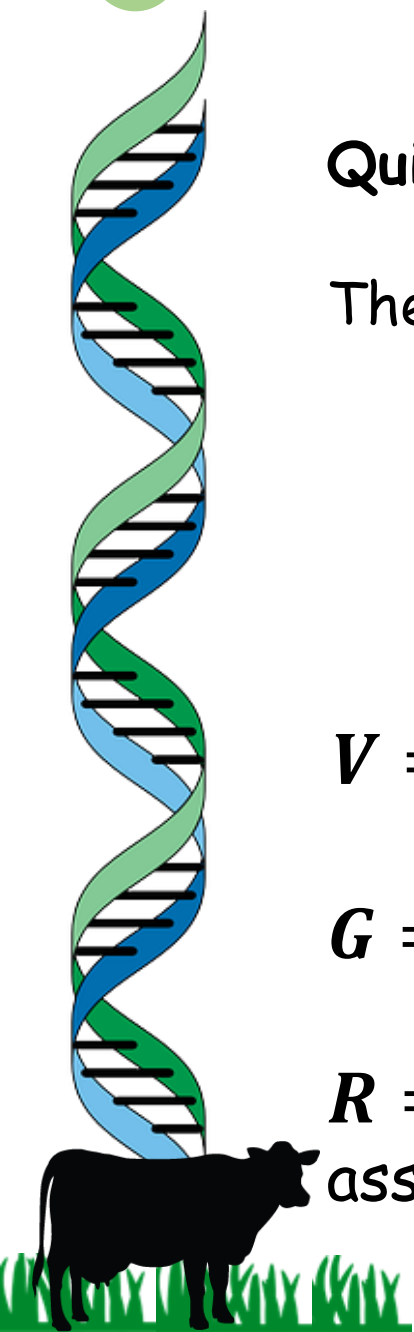
The animal model has the following distribution assumptions:

$$\begin{bmatrix} y \\ a \\ e \end{bmatrix} \sim N \left\{ \begin{bmatrix} X\beta \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} V & ZG & R \\ GZ^T & G & 0 \\ R & 0 & R \end{bmatrix} \right\}$$

$$V = ZGZ^T + R$$

$G = G_0 \otimes A$, which simplifies to $G = A\sigma_a^2$ for the univariate case

$R = R_0 \otimes \Sigma$, which simplifies to $R = I_n\sigma_e^2$ for the univariate case (and assuming independent residuals)



Quick review on mixed model equations derivation

The probability density function (p.d.f) of y is given by:

$$f(y) = \frac{1}{(2\pi^{N/2})|ZGZ^T + R|^{1/2}} \exp \left[-\frac{1}{2} (y - X\beta)^T (ZGZ^T + R)^{-1} (y - X\beta) \right]$$

Similarly, the joint p.d.f. of y and a can be written as:

$$f(y, a) = f(y | a) \cdot f(a)$$



Quick review on mixed model equations derivation

$$f(y, a) = \frac{1}{(2\pi^{N/2})|R|^{1/2}} \exp[-1/2 (y - X\beta - Za)^T R^{-1} (y - X\beta - Za)] \cdot \frac{1}{(2\pi^{N/2})|G|^{1/2}} \exp[-1/2 a^T G^{-1} a]$$

For mathematical convenience we can take the log of this function:

$$L = \frac{1}{2} 2N \log(2\pi) - \frac{1}{2} (\log|R| + \log|G|) - \frac{1}{2} [(y - X\beta - Za)^T R^{-1} (y - X\beta - Za) + a^T G^{-1} a]$$

$$= C - \frac{1}{2} (\log|R| + \log|G|) - \frac{1}{2} [y^T R^{-1} y - 2\beta^T X^T R^{-1} y - 2a^T Z^T R^{-1} y + 2\beta^T X^T R^{-1} Z a + \beta^T X^T R^{-1} X \beta + a^T Z^T G^{-1} Z a + a^T G^{-1} a]$$



Finally, by taking the derivatives of the log-likelihood with respect to β and a and setting them to zero we have:

$$\begin{bmatrix} \frac{dL}{d\beta} \\ \frac{dL}{da} \end{bmatrix} = \begin{bmatrix} X^T R^{-1} \mathbf{y} - X^T R^{-1} X \boldsymbol{\beta} - X^T R^{-1} Z \mathbf{a} \\ Z^T R^{-1} \mathbf{y} - Z^T R^{-1} X \boldsymbol{\beta} - Z^T R^{-1} Z \mathbf{a} - G^{-1} \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

$$\begin{bmatrix} X^T R^{-1} X \boldsymbol{\beta} + X^T R^{-1} Z \mathbf{a} \\ Z^T R^{-1} X \boldsymbol{\beta} + (Z^T R^{-1} Z + G^{-1}) \mathbf{a} \end{bmatrix} = \begin{bmatrix} X^T R^{-1} \mathbf{y} \\ Z^T R^{-1} \mathbf{y} \end{bmatrix}$$

$$\begin{bmatrix} X^T R^{-1} X & X^T R^{-1} Z \\ Z^T R^{-1} X & Z^T R^{-1} Z + G^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} X^T R^{-1} \mathbf{y} \\ Z^T R^{-1} \mathbf{y} \end{bmatrix}$$



One must notice that the MME application requires the knowledge of the (co)variances for the random effects. Those parameters are generally unknown and then are replaced by their estimates.

There are different algorithms for variance component estimation, among the most common in modern applications we have:

- a) DF-REML (Derivative-free REML)
- b) EM-REML (Expectation Maximization)
- c) PX-EM (Parameter Expanded EM)
- d) AIREML (Average Information - REML)
- e) Bayesian Inference via MCMC (Markov Chain Monte Carlo)



Expectation Maximization REML (EM-REML)

- ✓ The most stable algorithm (especially for complex models)
- ✓ Easy to implement
- ✓ May take hundreds of rounds of iteration to converge
- ✓ Does not calculate standard errors for the estimates



Expectation Maximization REML (EM-REML)

Considering the simplest case of the animal model:

$$y = X\beta + Za + e$$

The MME are:

$$\begin{bmatrix} X^T X & X^T Z \\ Z^T X & Z^T Z + A^{-1} \frac{\sigma_e^2}{\sigma_a^2} \end{bmatrix} \begin{bmatrix} \beta \\ a \end{bmatrix} = \begin{bmatrix} X^T y \\ Z^T y \end{bmatrix}$$

Further, let the inverse of the left-hand side be:

$$\begin{bmatrix} X^T X & X^T Z \\ Z^T X & Z^T Z + A^{-1} \frac{\sigma_e^2}{\sigma_a^2} \end{bmatrix}^{-1} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$



Expectation Maximization REML (EM-REML)

The EM algorithm can be implemented as following:

$$\begin{bmatrix} \hat{\beta}_{[i-1]} \\ \hat{a}_{[i-1]} \end{bmatrix} = \begin{bmatrix} X^T X & X^T Z \\ Z^T X & Z^T Z + A^{-1} \frac{\hat{\sigma}_e^{2[i-1]}}{\hat{\sigma}_a^{2[i-1]}} \end{bmatrix}^{-1} \begin{bmatrix} X^T y \\ Z^T y \end{bmatrix}$$

$$\hat{\sigma}_e^{2[i]} = \frac{e^T y}{N - \text{rank}(X)}$$

$$\hat{\sigma}_e^{2[i]} = \frac{y^T y - \hat{\beta}_{[i-1]}^T X^T y - \hat{a}_{[i-1]}^T Z^T y}{N - \text{rank}(X)}$$

$$\hat{\sigma}_a^{2[i]} = \frac{\hat{a}_{[i-1]}^T A^{-1} \hat{a}_{[i-1]} + \text{tr}(A^{-1} C_{22}) \hat{\sigma}_e^{2[i]}}{q}$$



The mixed model can be extended to accommodate different random effects, for example:

$$y = X\beta + Z_1a + Z_2m + Z_3pe + e$$



Additive genetic effect



Maternal genetic effect



Maternal permanent environmental effect

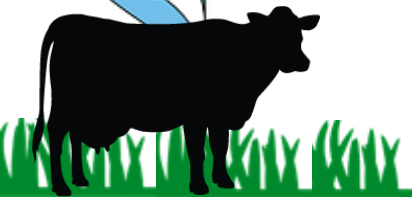


$$Y = X\beta + Za + e$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} X_1 & \mathbf{0} \\ \mathbf{0} & X_2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} Z_1 & \mathbf{0} \\ \mathbf{0} & Z_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

$$\begin{bmatrix} a \\ e \end{bmatrix} \sim N \left\{ \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} G_0 \otimes A & \mathbf{0} \\ \mathbf{0} & R_0 \otimes I \end{bmatrix} \right\}$$

$$G_0 = \begin{bmatrix} \sigma_{a11}^2 & \sigma_{a12} \\ \sigma_{a21} & \sigma_{a22}^2 \end{bmatrix} \text{ and } R_0 = \begin{bmatrix} \sigma_{e11}^2 & \sigma_{e12} \\ \sigma_{e21} & \sigma_{e22}^2 \end{bmatrix}$$



R packages

RRBLUP
PEDIGREEMM
BGLR
sommer
GeneticsPed
EMMREML
MCMCglmm



Easier implementation
Easier to adapt (open source)
User friendly
Memory-bounded

Popular software in animal breeding for dealing with huge data

ASREML <https://www.vsnr.co.uk/software/asreml/>
BLUPF90 <http://nce.ads.uga.edu/software/>
DMU <https://www.researchgate.net/project/DMU-software>
WOMBAT <http://didgeridoo.une.edu.au/km/homepage.php>

Generally written in Fortran 90/95
Flexible to deal with huge problems
Present less memory-related issues
Require parameter cards
Commercial license may apply



Data Simulation in R

Let's simulate the following trait: $y_{ijk} = \mu + \text{sex}_i + \text{protein}\%_j + a_k + e_{ijk}$

With the following features:

$\mu = 294$; *sex effect* = [25, -25]; *protein%* slope = 0.75; $\sigma_e^2 = 490$; $h^2 = 0.30$

$$\sigma_a^2 = h^2 \sigma_e^2 / (1 - h^2)$$

$a \sim N(0, \sigma_a^2)$ in the base population

$$a_k = \frac{1}{2} a_s + \frac{1}{2} a_d + \delta_k;$$

$$\delta_k \sim N\left(0, \frac{1}{2} \left\{1 - \frac{1}{2}(f_s + f_d)\right\} \sigma_a^2\right)$$

$a \sim N(0, A\sigma_a^2)$ in the expanded population

Further, let's assume the percentages of selected males and females (based on their phenotypic performance) as 10% and 50% at each generation.



Writing a program for VCE in R based on the EM-REML algorithm

Remember:

$$\begin{bmatrix} \hat{\beta}_{[i-1]} \\ \hat{a}_{[i-1]} \end{bmatrix} = \begin{bmatrix} X^T X & X^T Z \\ Z^T X & Z^T Z + A^{-1} \frac{\hat{\sigma}_e^{2[i-1]}}{\hat{\sigma}_a^{2[i-1]}} \end{bmatrix}^{-1} \begin{bmatrix} X^T y \\ Z^T y \end{bmatrix}$$

$$\hat{\sigma}_e^{2[i]} = \frac{e^T y}{N - \text{rank}(X)}$$

$$\hat{\sigma}_e^{2[i]} = \frac{y^T y - \hat{\beta}_{[i-1]}^T X^T y - \hat{a}_{[i-1]}^T Z^T y}{N - \text{rank}(X)}$$

$$\hat{\sigma}_a^{2[i]} = \frac{\hat{a}_{[i-1]}^T A^{-1} \hat{a}_{[i-1]} + \text{tr}(A^{-1} C_{22}) \hat{\sigma}_e^{2[i]}}{q}$$





mixedmod_tuto.R

Analyze the simulated data with the following packages/functions

- a) `emreml610` (EM-REML)
- b) `EMMREML` (EMMA-REML)
- c) `rrBLUP` (EM-ML)

Data sets
in tutorials



Data sets in
the wild

