In [ ]:

A linear mixed model in model organism association mapping is typically expressed as
$y = x\beta + Zu + e$
where y is an n × 1 vector of observed phenotypes, and x is an n × q matrix of fixed effects including mean, SNPs, and other confounding variables. β is a q × 1 vector representing coefficients of the fixed effects. Z is an n × t incidence matrix mapping each observed phenotype to one of t inbred strains. And in general Z is None, and will be ignored at some positons in the fellowing document. u is the random effect of the mixed model with $Var(u) = \sigma_g^2 K$, where K is the t × t kinship matrix, and e is an n × n matrix of residual effect such that $Var(e) = \sigma_e^2$. The overall phenotypic variance–covariance matrix can be represented as $V = \sigma_g^2 K + \sigma_e^2 I$
This part, in italic font, explains likelihood distribition and the symbols have no relationship with context.
For liner model y = xβ. The distribution of y $N(X\beta, \sigma^2 I)$ *So the likelihood* $L(y; \beta, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{(y-x\beta)'(y-x\beta)}{2\sigma^2}}$

For the linear mixed model: $y\ N(x\beta, \sigma_g^2 K + \sigma_e^2 I) = N(x\beta, \sigma_g^2 H)$
$H = \sigma_g^{-2} V = K + \delta I$ (there is a typo in the emma paper)
$\delta = \frac{\sigma_e^2}{\sigma_g^2}$
$likelihood(y; \beta, \sigma_g, \delta) = \frac{1}{(2\pi\sigma_g^2 H)^{\frac{n}{2}}} e^{-\frac{(y-x\beta)'H^{-1}(y-x\beta)}{\sigma_g^2}}$
ML (maximum likelihood):
$l_F(y; \beta, \sigma_g, \delta) = \frac{1}{2}(-nlog(2\pi\sigma_g^2) - log|H| - \frac{1}{\sigma_g^2}(y - x\beta)'H^{-1}(y - x\beta))$
REML: $l_R(y; \sigma_g, \delta) = l_F(y; \hat{\beta}, \sigma_g, \delta) + \frac{1}{2}(qlog(2\pi\sigma_g^2) + log|x'x| - log|x'H^{-1}x|)$
Gradient of the LMM log likelihood w.r.t. β
$\nabla_\beta l_R(y; \sigma_g, \delta) = \frac{d - \frac{1}{2\sigma_g^2}(y-x\beta)^T(K+I\delta)^{-1}(y-x\beta)}{d\beta} = \frac{1}{\sigma_g^2}(-x^T(k + I\delta)^{-1}y + x^T(k + I\delta)^{-1}x)$
set gradient to zero:
$x^T H^{-1} x\beta = x^T H^{-1} y$
β could be estimated as $\hat{\beta} = (x'H^{-1}x)^{-1}x'H^{-1}y$

Note that this solution is analogous to the ML solution of the linear regression
$(x^t x)^{-1}x^T y$

For ML: $\hat{\sigma}_g^2 = \frac{R}{n}$
Then: $l_F(y; \hat{\beta}, \hat{\sigma_g}, \hat{\delta}) = \frac{1}{2}(-nlog\frac{2\pi R}{n} - log|H| - n)$

For REML: $\hat{\sigma}_g^2 = \frac{R}{n-q}$
$R = (y - x\beta)'H^{-1}(y - x\beta)$
$H = K + \delta I = U_F(\xi_i + \delta, \ldots, \xi_n + \delta)U_F{}'$ with eigen decomposition
$K = U_F \xi U_F{}'$
so $log|H| = \sum_{i=1}^{n} log(\xi_i + \delta)$
And $R = (y - x\beta)'H^{-1}(y - x\beta) = y'(I - x(x'H^{-1}x)^{-1}x'H^{-1})'H^{-1}(I - x(x'H^{-1}x)^{-1}x'H^{-1})y = y'P'H^{-1}Py$

P is defined as $P = I - x(x'H^{-1}x)^{-1}x'H^{-1}$
$S = I - X(X'X)^{-1}X'$
$SHS = S(K + \delta I)S$ do eigen decomposition
And $(SHS)(P'H^{-1}P)(SHS) = SHS$
$(P'H^{-1}P)(SHS)(P'H^{-1}P) = (P'H^{-1}P)$
$PS = P$ (there is a typo in the emma paper)

and $SP = S$
$SHS = [U_R, W_R]diag(\lambda_1 + \delta, \ldots, \lambda_{n-q} + \delta, 0, \ldots, 0)[U_R, W_R]'$
$= U_R diag(\lambda_1 + \delta, \ldots, \lambda_{n-q}\delta, 0, \ldots, 0){U_R}'$

U_R is an n × (n – q) eigenvector matrix corresponding to the nonzero eigenvalues. W_R is an n × q eigenvector matrix corresponding to zero eigenvalues.
Here the eigen decomposition of H and SHS do not dependent on any unknow parameter and could be done directly

so $P'H^{-1}P = (SHS)^+ = U_R diag((\lambda_s + \delta)^{-1}){U_R}'$ here (.)^+ denotes the pseudo-inverse of a matrix Let
${U_R}'y = [\eta_1, \eta_2, \ldots, \eta_{n-q}]'$
and $R = y'P'H^{-1}Py = ({U_R}'y)'diag((\lambda_s + \delta)^{-1})({U_R}'y) = \sum_{s=1}^{n-q} \frac{\eta_s^2}{\lambda_s+\delta}$

Then for ML: $l_F(y; \hat{\beta}, \hat{\sigma_g}, \hat{\delta}) = \frac{1}{2}nlog\frac{n}{2\pi} - n - nlog\sum_{s=1}^{n-q}\frac{\eta_s^2}{\lambda_s+\delta} - \sum_{i=1}^{n} log(\xi_i + \delta)$

$(SHS)(SHS)^+ = (SHS)(P'H^{-1}P) = SHP'H^{-1}P = SP = S$
On the other hand
$(SHS)(SHS)^+ = (U_R diag(\lambda_s + \delta){U_R}')(U_R diag((\lambda_s + \delta)^{-1}){U_R}') = U_R{U_R}'$
so $U_R{U_R}' = S = I$
taking account $\hat{\sigma}_g^2 = \frac{R}{n-q}$
Then for REML: $l_R(y; \hat{\sigma_g}, \hat{\delta}) = \frac{1}{2}(n-q)log\frac{n-q}{2\pi} - (n-q) - (n-q)log\sum_{s=1}^{n-q}\frac{\eta_s^2}{\lambda_s+\delta} - \sum_{s=1}^{n-q} log(\lambda_s + \delta)$

The derivatives of these functions:

ML: $f_F{}' = \frac{n}{2}\frac{\sum_s \frac{\eta_s^2}{(\lambda_s+\delta)^2}}{\sum_s \frac{\eta_s^2}{\lambda_s+\delta}} - \frac{1}{2}\sum_i \frac{1}{\xi_i+\delta}$

REML: $f_F{}' = \frac{n-q}{2}\frac{\sum_s \frac{\eta_s^2}{(\lambda_s+\delta)^2}}{\sum_s \frac{\eta_s^2}{\lambda_s+\delta}} - \frac{1}{2}\sum_i \frac{1}{\xi_i+\delta}$

if we could find B such that
$BB' = H = \frac{V}{\sigma_g^2} = K + \delta I$
we can substitete $y^* = B^{-1}y$, $x^* = B^{-1}x$ and $\epsilon^* = B^{-1}(Zu + \epsilon)$ (now ϵ^* includes both random effects and errors) to get
$y^* = x^*\beta + \epsilon^*$
$Var(\epsilon^*) = Var(B^{-1}(Zu + \epsilon)) = B^{-1}V(B^{-1})' = \sigma_g^2 B^{-1}H(B^{-1})' = \sigma_g^2 B^{-1}BB'(B^{-1})' = \sigma_g^2 I$
The value of the residual sum of squares (RSS) from solving the transformed equation $y^* = X^*\beta + \epsilon^*$ is the Mahalanobis RSS for the original equation $y = x\beta + Zu + \epsilon$.

Taking advantage of the eigen decomposition of H performed in the EMMA algorithm, the computation of a valid B^{-1} can be simplified to
$B^{-1} = diag(1/\sqrt{\xi_1 + \delta}, \ldots, 1/\sqrt{\xi_n + \delta})U_F'$

B^{-1} is H_sqrt_inv in the code

β could be estimate by solving $y^* = x^*\beta + \epsilon^*$

The F test here is performed on y^* and x^*
In the code h0_rss is the Residual sum of squares without the effecte of fixed variable under testing (H0)
mahalanobis_rss is the Residual sum of squares with the effecte of fixed variable under testing (H1)

If here x is the intercept and all the significant genotypic variants, after β being calculated
Then residuals = y - xβ
Variance explained by significant genotypic variants is (var(y)-residuals.T * residuals) / var(y)

In [ ]: