Simulation of multiple trait data for testing breeding value estimation programs

Institute of Animal Breeding and Husbandry with Veterinary Clinic, Martin-Luther-University of Halle-Wittenberg[1]
and Institute of Mathematics, University Potsdam, Germany[2]

M. WENSCH-DORENDORF[1], H.H. SWALVE[1] and J. WENSCH[2]

dorendorf@landw.uni-halle.de

Session G4.36
Abstract no. 219

Introduction

- computer programs to be used for routine genetic evaluations need to be tested and checked
- testing of such programs requires simulated data with known phenotypes and breeding values

The idea

- generate breeding values for base animals, derive for nonbase animals
- substitute the values in the mixed model equations and evaluate phenotypic data with these equations
- Thompson (1997): Proposal for single trait model

⇒ **the solutions to the mixed model equations yield the exact values of the simulated ones**

Method of simulation

- simulate randomized breeding values for base animals (u_b) with mean zero
- compute breeding values for nonbase animals (u_{nb}) from their respective parents by using $u_{nb}=0.5(u_s+u_d)$
- substitute breeding values in the mixed model equation and rearrange to compute phenotypic data
- the generated data satisfy the mixed model equations **exact**

Method of breeding value estimation

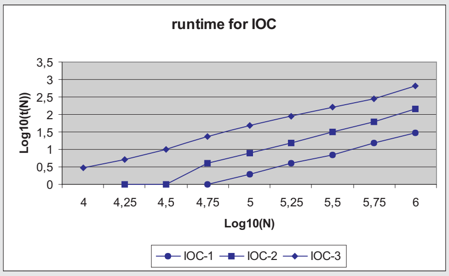
- a Fortran90 PEST-Version (Groeneveld) was used for breeding value estimation (single, two and three trait model)
- three solvers were tested:
  - -IOC: iterative solver using gauss-seidel-algorithm, all coefficients stored in memory (in memory iteration)
  - -IOD_GS: iterative solver using gauss-seidel-algorithm as iteration on data
  - -SMP: direct solver for sparse matrices

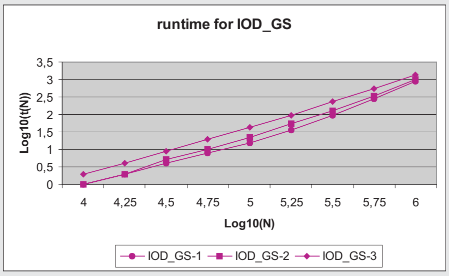
Results

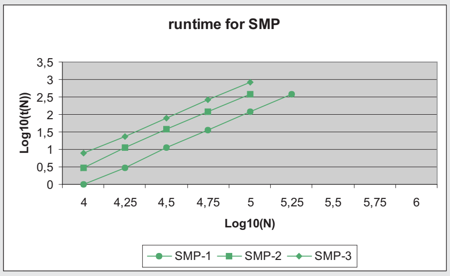
- datasets with N=10000, 17783, 31623, 56235, 100000, 177828, 316228, 562340, 1000000 animals were simulated with single, two and three trait model

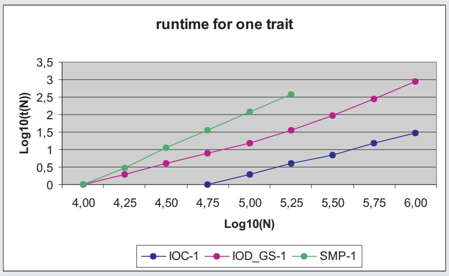
⇒ breeding value estimation with PEST

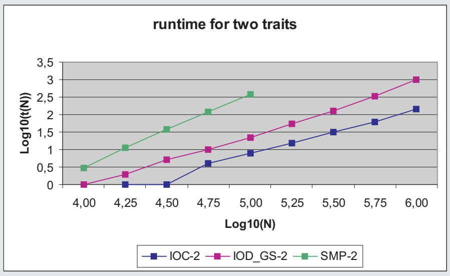
- runtime t was measured
- the figures show the (N, t) pairs (in Log10-scale) sorted by solvers and by number of traits (number 1, 2 or 3 after solver means single, two or three trait model)

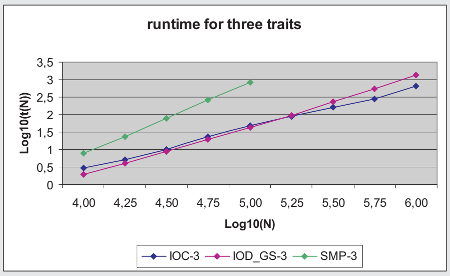












Conclusions

- the iterative solvers are faster than SMP - runtime for SMP grows quadratically (estimation of *a* from: *Log10(t)=b+a*Log10(N)* is approx. 2.0)
- runtimes for IOC and IOD_GS grow almost weakly linearly (estimations of *a* were mostly >1.0 and < 2.0), but the values of *a* for IOC solver were often smaller than for IOD_GS
- differences between simulated and estimated breeding values were very small - averages for differences were $1.4 \cdot 10^{-5}$ to $6 \cdot 10^{-6}$ for the small N`s and $4 \cdot 10^{-6}$ to $2 \cdot 10^{-6}$ for larger N`s

⇒ **the method is very suitable to simulate multiple trait phenotypic data from generated breeding values which can be estimated exact**

Derivation for multiple trait model

general mixed model equation
($R=R_0 \otimes I$, $G=G_0 \otimes A$, $G^{-1}=G_0^{-1} \otimes A^{-1}$)

⇒ **mixed model equation** (for Z=I) :

⇒ **simulation equation of phenotypic data** (for none base and base animals for traits *k=1,...,n*)

$$\begin{bmatrix} X^TR^{-1}X & X^TR^{-1}Z \\ Z^TR^{-1}X & Z^TR^{-1}Z+G^{-1} \end{bmatrix} \begin{bmatrix} \hat\beta \\ \hat u \end{bmatrix} = \begin{bmatrix} X^TR^{-1}y \\ Z^TR^{-1}y \end{bmatrix} \Rightarrow \begin{bmatrix} X^TR^{-1}X & X^TR^{-1} \\ R^{-1}X & R^{-1}+G^{-1} \end{bmatrix} \begin{bmatrix} \hat\beta \\ \hat u \end{bmatrix} = \begin{bmatrix} X^TR^{-1}y \\ R^{-1}y \end{bmatrix} \Rightarrow$$

$$\begin{aligned} y_{nbk} &= X\hat\beta + \hat u_{nbk} \\ y_{bk} &= X\hat\beta + \hat u_{bk} + \textstyle\sum_{i=1}^{n} R_k G^i \hat u_{bi} \end{aligned}$$

by using the following equations
(R_i=i-th row of R_0 and G^i=i-th column of G^{-1}_0)

$$A^{-1}\hat u_i = \begin{bmatrix} \hat u_{bi} \\ 0 \end{bmatrix}$$

$$R_0 = \begin{pmatrix} \sigma^2_{e_1} & \sigma_{e_1e_2} & \cdots & \sigma_{e_1e_n} \\ \sigma^2_{e_2e_1} & \sigma^2_{e_2} & \cdots & \sigma_{e_2e_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{e_ne_1} & \sigma_{e_ne_2} & \cdots & \sigma^2_{e_n} \end{pmatrix} \quad G_0 = \begin{pmatrix} \sigma^2_{a_1} & \sigma_{a_1a_2} & \cdots & \sigma_{a_1a_n} \\ \sigma^2_{a_2a_1} & \sigma^2_{a_2} & \cdots & \sigma_{a_2a_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{a_na_1} & \sigma_{a_na_2} & \cdots & \sigma^2_{a_n} \end{pmatrix}$$

$$R_0 = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{pmatrix} \quad G_0^{-1} = \begin{pmatrix} G^1 & G^2 & \cdots & G^n \end{pmatrix}$$

$$RG^{-1} = \begin{pmatrix} R_1G^1 & \cdots & R_1G^n \\ \vdots & \ddots & \vdots \\ R_nG^1 & \cdots & R_nG^n \end{pmatrix} \otimes A^{-1}$$

Householder reflection is used to simulate symmetric positive definite matrices: $S=Q\Lambda Q^T$, Q is an orthogonal matrix, Λ is a diagonal matrix with eigenvalues λ_i of S in the main diagonal; if all $\lambda_i>0$ then S is positive definite, using householder reflection to generate Q: $Q(v):=I-2(vv^T)/(v^Tv)$, v any vector with dimension n=number of traits

Starting values:

single trait: $G_0 = 53.0 \quad R_0 = 75$

two trait: $G_0 = \begin{pmatrix} 73.969 & 7.105 \\ 7.105 & 26.031 \end{pmatrix} \quad R_0 = \begin{pmatrix} 57.075 & -0.469 \\ -0.469 & 59.925 \end{pmatrix}$

three trait: $G_0 = \begin{pmatrix} 52.923 & -1.164 & 1.080 \\ -1.164 & 33.552 & 22.047 \\ 1.080 & 22.047 & 20.525 \end{pmatrix} \quad R_0 = \begin{pmatrix} 59.692 & -2.784 & -35.262 \\ -2.784 & 64.264 & 25.091 \\ -35.2622 & 25.091 & 40.044 \end{pmatrix}$