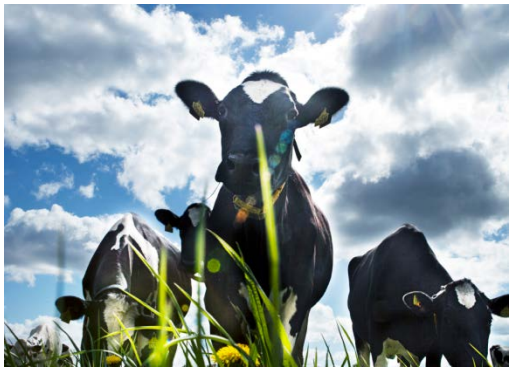
Efficient multi-trait SNP models for prediction of genomic breeding values

Luc Janss
Center for Quantitative Genetics and Genomics
Aarhus University, Denmark


SNP-BLUP and GBLUP

Meuwissen 2001:

- Genomic breeding value = Σ genotypes x SNP-effects
- SNP-effect is random regression on allele dosage -> "BLUP" of SNP effects
- Should be more efficient when nr individuals > markers

Van Raden 2008

- Can also compute directly GBV using covar(g) = genotypes x genotypes^T
  -> using G-matrix -> GBLUP
- Should be more efficient when nr individuals < markers

But comparison

- Does not consider convergence behavior
- Does not consider multi-trait
- Does not consider possible strategies to improve convergence
- Does not consider G-matrix preparation and inversion
- (Does not consider single-step)

Aims - Outline

- Improve convergence in SNP-BLUP using joint update of SNP-effects
- Improve convergence in multi-trait SNP-BLUP by decomposing trait-covariances
- Investigate computing time (multi-trait) SNP-BLUP vs GBLUP
- Further extensions

Material

- Simulated population under selection
- 2000 QTL, effects ~ MVN
- 40K (36K) SNPs
- 200’000 individuals – using last 40’000

GBLUP

- Iteration-on-data (DMU5) using Preconditioned Conjugate Gradient
- G-matrices computed and inverted using DMU “G-matrix”

- Comparisons excluding parallelisation
- Convergence on $\text{norm}(g_i - g_{i-1})/\text{norm}(g_i) < 10^{-8}$

SNP effect joint updating

$$y = Xb + w_1 a_1 + w_2 a_2 + \ldots + w_{40K} + e$$

$$y - Xb - \sum_{i \neq j} w_i a_i = w_j a_j + e$$

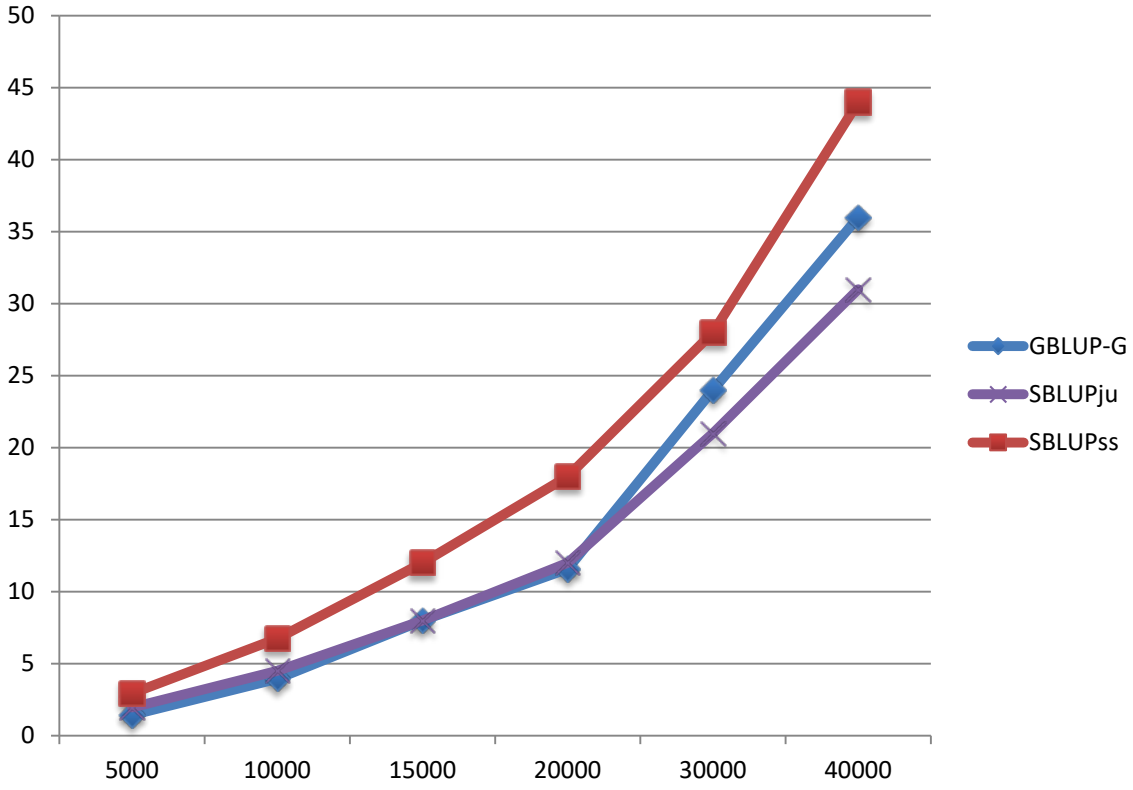
- Update 1 SNP effect at a time
  - Potentially slow convergence when high LD

$$y - Xb - \sum_{i \neq j,k,l} w_i a_i = w_j a_j + w_k a_k + w_l a_l + e$$

- Update several jointly
  - Do quickly by absorption in MME
  - 5 SNPs: no extra computing time
  - 10 SNPs: some extra computing time, but still worth the speed-up
- Applied in sliding windows on physical ordering of SNPs

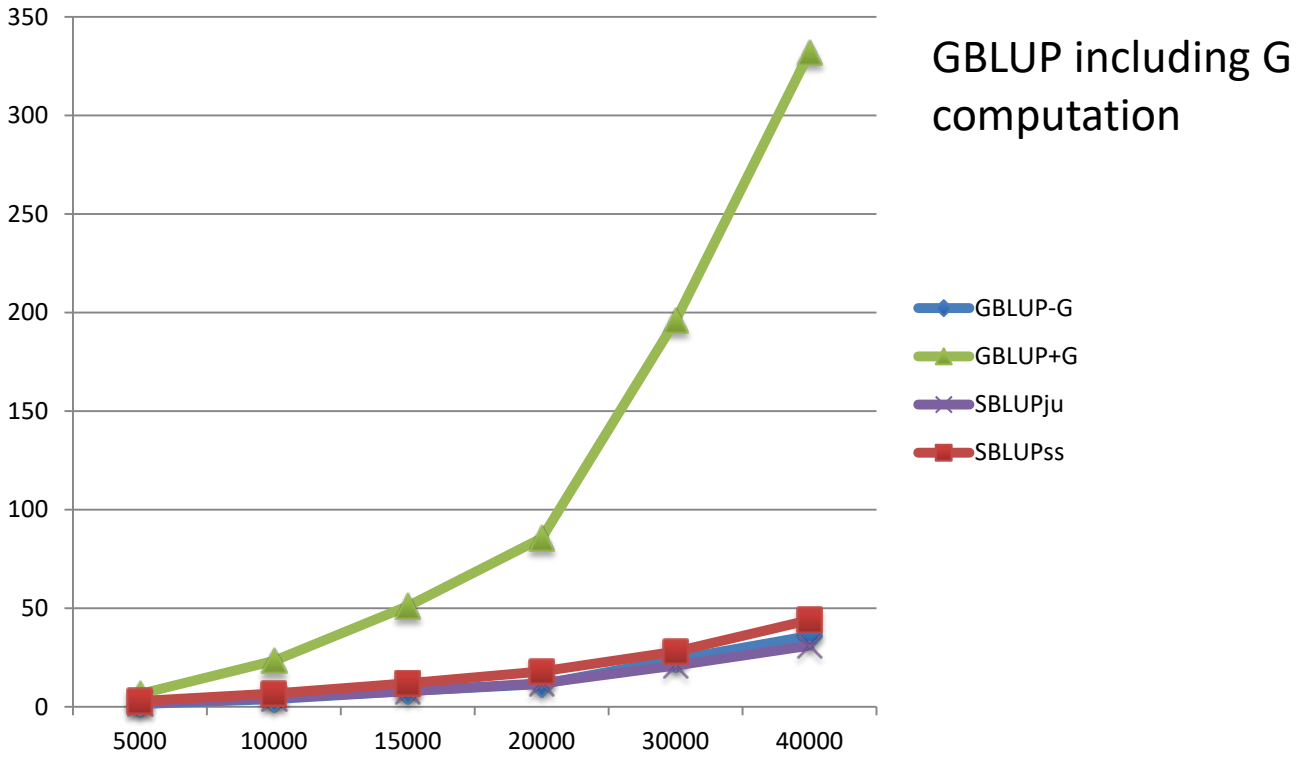
Single trait comparison:

computing time (min) for SBLUP single-snp updating, SBLUP joint updating and GBLUP (excl G computation)



Single trait comparison:

computing time (min) for SBLUP single-snp updating, SBLUP joint updating and GBLUP (excl G computation)



Multi-trait covariance decomposition

- Spectral decomposition decomposes covariance in sum of independent parts

$$R_{3x3} = \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \lambda_3 v_3 v_3^T$$

λ_i eigenvalues, v_i eigen-vectors

- The last eigenvector is determined by all other eigenvectors, in general for K traits

$$v_K v_K^T = I - \sum_{i=1}^{i<K} v_i v_i^T$$

- And the spectral decomposition can be rewritten:

$$R = \tilde{\lambda}_1 v_1 v_1^T + \tilde{\lambda}_2 v_2 v_2^T + I\lambda_3$$

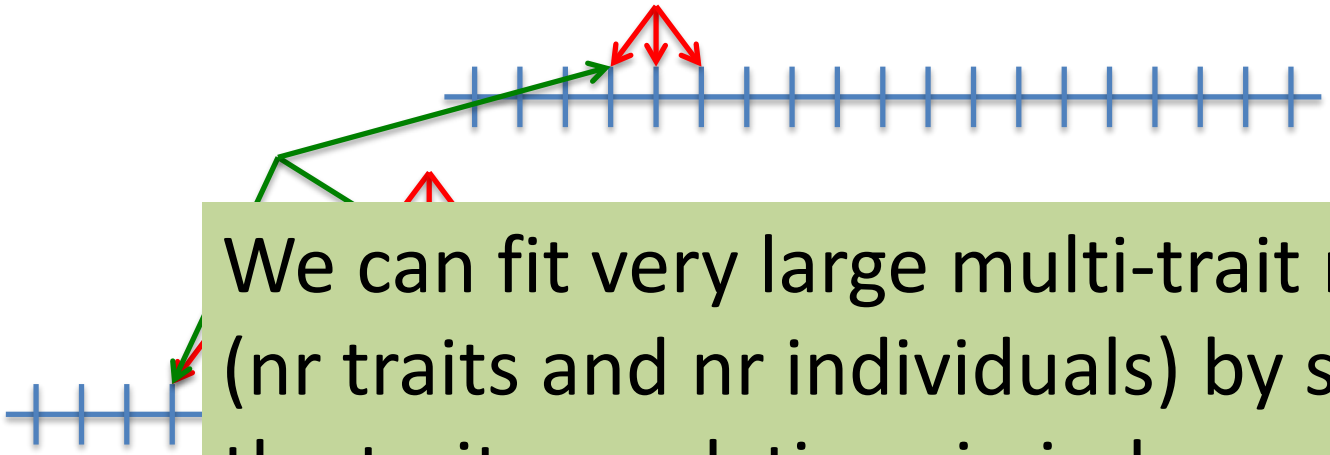
- This allows to fit the covariance by introducing random effects across traits x regression 'loadings'

$$\begin{cases} y_1 = X_1 b_1 + W a_1 + e_1 & a_1 \sim N(v_{11} s_1 + v_{21} s_2,\ I\lambda_3) \\ y_1 = X_2 b_2 + W a_2 + e_2 & a_2 \sim N(v_{12} s_1 + v_{22} s_2,\ I\lambda_3) \\ y_1 = X_3 b_3 + W a_3 + e_3 & a_3 \sim N(v_{13} s_1 + v_{23} s_2,\ I\lambda_3) \end{cases}$$

$$\text{var}(s_1) = I\tilde{\lambda}_1$$

$$\text{var}(s_2) = I\tilde{\lambda}_2$$

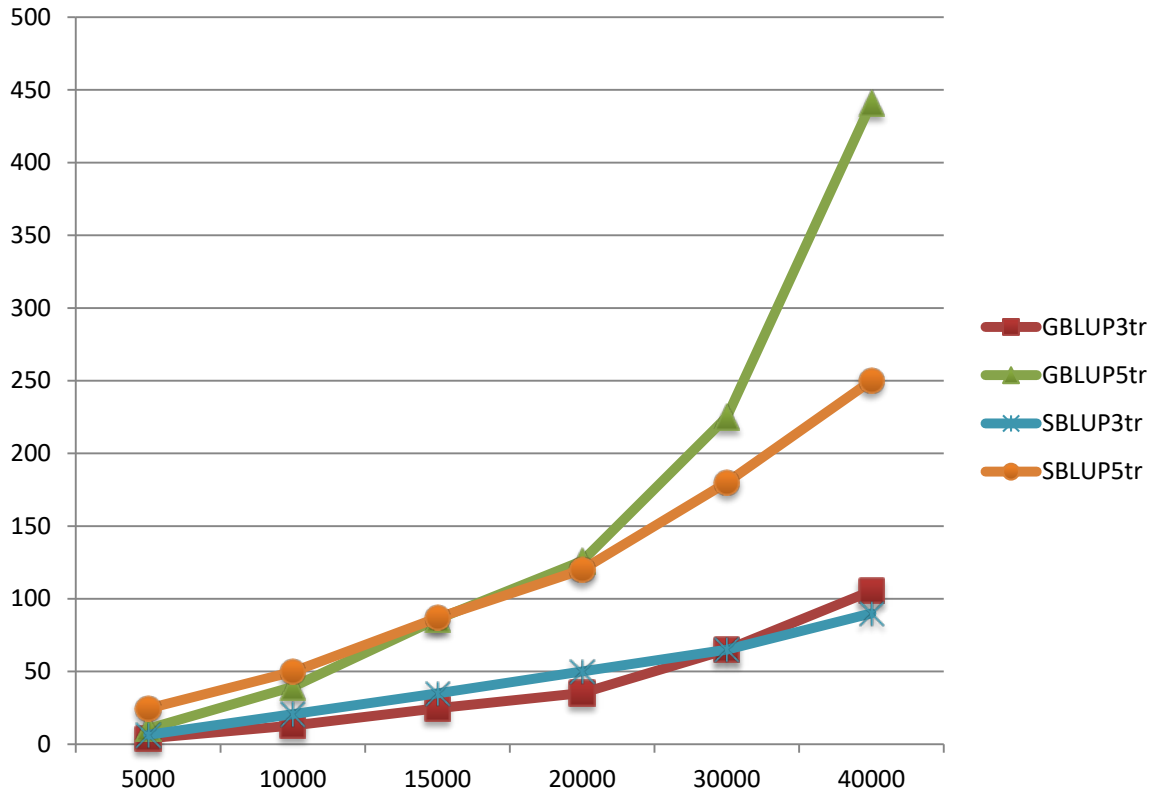
Integrating ….



We can fit very large multi-trait models (nr traits and nr individuals) by splitting the trait correlations in independent parts suitable for iterative computation of breeding values

3-trait and 5-trait with GBLUP and SBLUP

(GBLUP excl G computation)



Scaling up summary

- Doubling individuals
  - 4x increase in computing + inverting G
  - 2.5 – 3.5 increase in computing GEBV

- Increasing number of traits
  - 3-trait analysis ≈ 3x unitrait for S-BLUP and GBLUP
  - 5-trait analysis ≈ 5x unitrait for S-BLUP
    ≈ 8-12x unitrait for GBLUP

Discussion - Conclusions

- Simple logic that GBLUP faster when individuals < markers does not hold
  - Not-optimised SBLUP seems not to catch up
  - Optimised SBLUP gets faster before indiv > markers
  - Multi-trait SBLUP gets faster quicker when traits >3
- G-matrix preparation is significant task
  - Multi-trait using one G 'cost' of computing G spreads over multiple traits
  - Multi-trait weighted SNP model 'cost' of computing G increases by need for all trait x trait weighted G's
- Parallelisation possible
  - Main tendencies remain unless one is better parallelisable than the other
- Approximations possible
  - Large (single-step) GBLUP already needs approximations!
  - SBLUP can use reduced rank SNPs and covariances