Efficient computations for single-step genomic evaluations

I. Strandén,
K. Matilainen, G.P. Aamand, E.A. Mäntysaari

Luke
NATURAL RESOURCES INSTITUTE FINLAND

Introduction

- Single-step MME: $\begin{bmatrix} \mathbf{X'X} & \mathbf{X'W} \\ \mathbf{W'X} & \mathbf{W'W} + \lambda \mathbf{H}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'y} \\ \mathbf{Z'y} \end{bmatrix}$

where $\lambda = \frac{\sigma_e^2}{\sigma_a^2}$ and $\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{A}^{11} & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \mathbf{A}^{22} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - (\mathbf{A}_{22})^{-1} \end{bmatrix}$

- Number of genotype animals increases
  - ➔Making and inverting $\mathbf{A}_{22}$ and **G** matrices becomes difficult
- This study:
  - APY (Algorithm for Proven and Young animals) by Cholesky
  - Matrix vector product $\mathbf{A}_{22}^{-1}\mathbf{d}_2$ without ever making the $\mathbf{A}_{22}$ matrix
    - Three different approaches
  - Application: Nordic Holstein fertility model data

Original APY (Misztal et al., 2015)

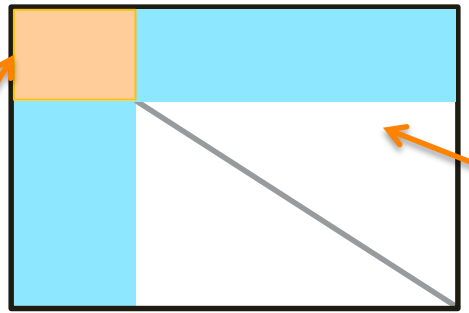
APY idea:

- Divide to core (c) and young (y) animals
- Approximate **G** inverse by making $\mathbf{G}^{yy}$ diagonal

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{cc} & \mathbf{G}_{cy} \\ \mathbf{G}_{yc} & \mathbf{G}_{yy} \end{bmatrix}$$

$$\mathbf{G}^{-1} = \begin{bmatrix} \mathbf{G}^{cc} & \mathbf{G}^{cy} \\ \mathbf{G}^{yc} & \mathbf{G}^{yy} \end{bmatrix}$$

$$\mathbf{G}_{\text{APY}}^{-1} = \begin{bmatrix} \mathbf{G}_{cc}^{-1} + \mathbf{G}_{cc}^{-1}\mathbf{G}_{cy}\mathbf{M}_{yy}^{-1}\mathbf{G}_{yc}\mathbf{G}_{cc}^{-1} & -\mathbf{G}_{cc}^{-1}\mathbf{G}_{cy}\mathbf{M}_{yy}^{-1} \\ -\mathbf{M}_{yy}^{-1}\mathbf{G}_{yc}\mathbf{G}_{cc}^{-1} & \mathbf{M}_{yy}^{-1} \end{bmatrix}$$



$\mathbf{M}_{yy}$= Diagonal of $\mathbf{G}_{yy} - \mathbf{G}_{yc}\mathbf{G}_{cc}^{-1}\mathbf{G}_{cy}$

Large sparse sub-matrix
→ Lower storage need
→ Lower computational load

Number of core animals low: less than number of markers

Original APY (Misztal et al., 2015)

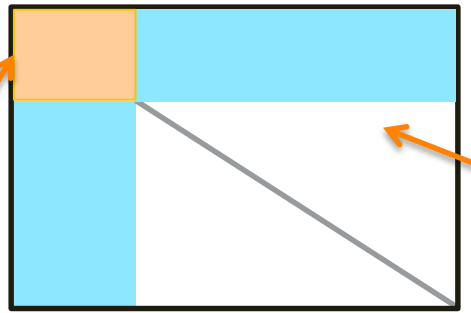
APY idea:

- Divide to core (c) and young (y) animals
- Approximate **G** inverse by making $\mathbf{G}^{yy}$ diagonal

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{cc} & \mathbf{G}_{cy} \\ \mathbf{G}_{yc} & \mathbf{G}_{yy} \end{bmatrix}$$

$$\mathbf{G}^{-1} = \begin{bmatrix} \mathbf{G}^{cc} & \mathbf{G}^{cy} \\ \mathbf{G}^{yc} & \mathbf{G}^{yy} \end{bmatrix}$$

$$\mathbf{G}_{\text{APY}}^{-1} = \begin{bmatrix} \mathbf{G}_{cc}^{-1} + \mathbf{G}_{cc}^{-1}\mathbf{G}_{cy}\mathbf{M}_{yy}^{-1}\mathbf{G}_{yc}\mathbf{G}_{cc}^{-1} & -\mathbf{G}_{cc}^{-1}\mathbf{G}_{cy}\mathbf{M}_{yy}^{-1} \\ -\mathbf{M}_{yy}^{-1}\mathbf{G}_{yc}\mathbf{G}_{cc}^{-1} & \mathbf{M}_{yy}^{-1} \end{bmatrix}$$



$$\mathbf{G}_{\text{APY}} = \begin{bmatrix} \mathbf{G}_{cc} & \mathbf{G}_{cy} \\ \mathbf{G}_{yc} & \mathbf{M}_{yy} + \mathbf{G}_{yc}\mathbf{G}_{cc}^{-1}\mathbf{G}_{cy} \end{bmatrix}$$

Large sparse sub-matrix
→ Lower storage need
→ Lower computational load

Number of core animals low: less than number of markers

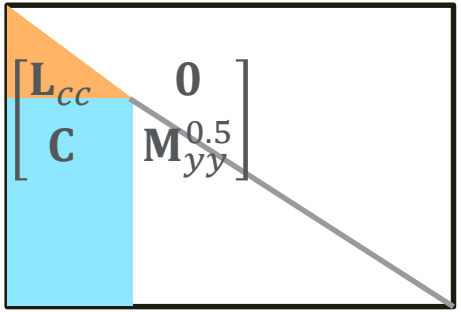
APY by Cholesky decomposition

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{cc} & \mathbf{G}_{cy} \\ \mathbf{G}_{yc} & \mathbf{G}_{yy} \end{bmatrix}$$

$$\mathbf{G}_{\mathrm{APY}} = \begin{bmatrix} \mathbf{G}_{cc} & \mathbf{G}_{cy} \\ \mathbf{G}_{yc} & \mathbf{M}_{yy} + \mathbf{G}_{yc}\mathbf{G}_{cc}^{-1}\mathbf{G}_{cy} \end{bmatrix}$$

Steps:

1. Let **L** in $\mathbf{G}_{apy} = \mathbf{L}\,\mathbf{L}'$ where $\mathbf{L} = \begin{bmatrix} \mathbf{L}_{cc} & \mathbf{0} \\ \mathbf{C} & \mathbf{M}_{yy}^{0.5} \end{bmatrix}$



where:

- $\mathbf{G}_{cc} = \mathbf{L}_{cc}\,\mathbf{L}_{cc}'$
- $\mathbf{C} = \mathbf{G}_{yc}\,\mathbf{L}_{cc}^{-1}$
- $\mathbf{M}_{yy}$ = diagonal of $\mathbf{G}_{yy}$-$\mathbf{CC}'$

2. Calculate the inverse

$$\mathbf{G}_{\mathrm{APY}}^{-1} = \begin{bmatrix} (\mathbf{L}_{cc}')^{-1}\mathbf{L}_{cc}^{-1} + \mathbf{B}'\mathbf{B} & -\mathbf{B}\mathbf{M}_{yy}^{-0.5} \\ -\mathbf{M}_{yy}^{-0.5}\mathbf{B} & \mathbf{M}_{yy}^{-1} \end{bmatrix}$$

where $\mathbf{B} = \mathbf{M}_{yy}^{-0.5}\mathbf{C}\mathbf{L}_{cc}^{-1}$

APY original vs. Cholesky approach

Original:

$$\mathbf{G}_{\text{APY}}^{-1} = \begin{bmatrix} \mathbf{G}_{cc}^{-1} + \mathbf{G}_{cc}^{-1}\mathbf{G}_{cy}\mathbf{M}_{yy}^{-1}\mathbf{G}_{yc}\mathbf{G}_{cc}^{-1} & -\mathbf{G}_{cc}^{-1}\mathbf{G}_{cy}\mathbf{M}_{yy}^{-1} \\ -\mathbf{M}_{yy}^{-1}\mathbf{G}_{yc}\mathbf{G}_{cc}^{-1} & \mathbf{M}_{yy}^{-1} \end{bmatrix}$$

$\mathbf{M}_{yy}$= Diagonal of $\mathbf{G}_{yy} - \mathbf{G}_{yc}\mathbf{G}_{cc}^{-1}\mathbf{G}_{cy}$

Using Cholesky:

$$\mathbf{G}_{\text{APY}}^{-1} = \begin{bmatrix} (\mathbf{L}'_{cc})^{-1}\mathbf{L}_{cc}^{-1} + \mathbf{B}'\mathbf{B} & -\mathbf{B}\mathbf{M}_{yy}^{-0.5} \\ -\mathbf{M}_{yy}^{-0.5}\mathbf{B} & \mathbf{M}_{yy}^{-1} \end{bmatrix}$$

where

$$\mathbf{B} = \mathbf{M}_{yy}^{-0.5}\mathbf{C}\mathbf{L}_{cc}^{-1} \qquad \mathbf{C} = \mathbf{G}_{yc}(\mathbf{L}'_{cc})^{-1} \qquad \mathbf{M}_{yy} = \text{Diagonal of } \mathbf{G}_{yy} - \mathbf{C}\mathbf{C}'$$

Cholesky decomposition brings
- numerical stability through decomposition
- programming simplicity with (parallel) BLAS and LAPACK subroutines

A_{22} inverse part

- Matrix blocks:
- Use equality:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \qquad \mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}^{11} & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \mathbf{A}^{22} \end{bmatrix}$$

$$\left(\mathbf{A}_{22}\right)^{-1} = \mathbf{A}^{22} - \mathbf{A}^{21}\left(\mathbf{A}^{11}\right)^{-1}\mathbf{A}^{12}$$

- Steps to calculate $\mathbf{z}_2 = (\mathbf{A}^{22} - \mathbf{A}^{21}\,(\mathbf{A}^{11})^{-1}\,\mathbf{A}^{12}\,)\,\mathbf{d}_2$
  1. $[\,\mathbf{x}_1\ \mathbf{x}_2]' = [\mathbf{A}^{21}\ \mathbf{A}^{22}]'\ \mathbf{d}_2$
  2. $\mathbf{y}_1 = (\mathbf{A}^{11})^{-1}\,\mathbf{x}_1$
  3. $\mathbf{z}_2 = \mathbf{x}_1 - \mathbf{A}^{21}\,\mathbf{y}_1$

Pedigree read: $\mathbf{A}^{-1}$ rules

- Step 2. alternatives:

IOP: PCG solve by **Iteration on Pedigree**: pedigree reading

IM: PCG solve by $\mathbf{A}^{11}$ **in memory**

CM: Direct solving using **CHOLMOD** library

Study design

$$\begin{bmatrix} \mathbf{X'X} & \mathbf{X'W} \\ \mathbf{W'X} & \mathbf{W'W} + \lambda \mathbf{H}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'y} \\ \mathbf{Z'y} \end{bmatrix}$$

Models:

- AMBLUP: Animal model BLUP, no genomics
- ssGBLUP: single-step, has genomic information

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{A}^{11} & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \mathbf{A}^{22} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - (\mathbf{A}_{22})^{-1} \end{bmatrix}$$

Inverse of $\mathbf{A}_{22}$

- Calculate $(\mathbf{A}^{11})^{-1}$ → **regular ssGBLUP**
- Step 2: $\mathbf{y}_1 = (\mathbf{A}^{11})^{-1}\,\mathbf{x}_1$
  - PCG and IOP: **iteration on pedigree**
  - PCG and IM: $\mathbf{A}^{11}$ **in memory**
  - Direct: **CHOLMOD**

APY 10,000 or 20,000 in core: Highest number of progeny

Luke
NATURAL RESOURCES INSTITUTE FINLAND

Holstein fertility data and Eurogenomics genotyped animals

- 81,031 genotyped (clones removed)
  - 46,344 markers
  - APY: core was 10,000 (10K) or 20,000 (20K) animals
- 9.73 million animals in pedigree
- 7.5 million record animals
  - 11 traits, fertility model, no genetic groups
  - 48.7 million observations

File sizes:

- Text genotype marker data: 7 GB
- Binary $\mathbf{G}^{-1}$ – $(\mathbf{A}_{22})^{-1}$ and $\mathbf{G}^{-1}$ file sizes: 13 GB
- Binary APY $\mathbf{G}^{-1}$ file size: 2.9 GB (10K), 5.3 GB (20K)

First results: ssGBLUP: APY and A_{22}^{-1} by IOP

		Time	Total	N iter
AMBLUP	Preprocessor	6m		
	Solver	6h 15m	6h 21m	1956
ssGBLUP	A_{22} by RelaX2	2h 48m		
regular	$\mathbf{G}^{-1} - (\mathbf{A}_{22})^{-1}$	24h 26m		
	Preprocessor	16m		
	Solver	38h 2m	65h 32m	2088
ssGBLUP	$\mathbf{G}^{-1}$	10h 19m		
No A_{22}^{-1} build	Preprocessor	17m		
IOP	Solver	52h 12m	62h 48m	2108
ssGBLUP	$\mathbf{G}^{-1}$	5h 22m		
APY 10K	Preprocessor	9m		
	Solver	28h 13m	34h 44m	1963

Parallel computing would allow faster computations especially for precalculating matrices like $\mathbf{G}^{-1}$ and $\mathbf{G}^{-1} - (\mathbf{A}_{22})^{-1}$

Luke
NATURAL RESOURCES INSTITUTE FINLAND

Solver CPU time and peak memory

	Mem (GB)	Solver time	Time/iter	N iter
AMBLUP	3.61	**8h**	0.24m	1981
Regular ssGBLUP	3.62	36h	1.04m	2059
No A_{22}^{-1} build, IOP	4.61	49h	1.44m	2038
No A_{22}^{-1} build, IM	4.62	42h	1.22m	2046
No A_{22}^{-1} build, CM	4.78	36h	1.06m	2041
APY 20K, CM	4.78	21h	0.60m	2049
APY 10K, CM	4.78	15h	0.46m	1915

Regular ssGBLUP: Additional time due to A_{22} is about 14 hours (building and inversion).

Correlation and std for GEBV of genotyped animals: ssGBLUP vs. APY

10,000 in core

Trait	Corr	Std ssGBLUP	Std APY
NRR0	0.999	2.79	2.78
IFL0	0.999	3.32	3.31
NRR1	0.999	4.25	4.24
ICF1	0.999	6.56	6.55
IFL1	0.999	10.28	10.27
NRR2	0.999	4.36	4.33
ICF2	0.999	6.85	6.85
IFL2	0.999	10.05	10.03
NRR3	0.999	4.04	4.02
ICF3	0.999	6.57	6.56
IFL3	0.999	9.38	9.33

20,000 in core

Trait	Corr	Std ssGBLUP	Std APY
NRR0	1.000	2.79	2.78
IFL0	1.000	3.32	3.32
NRR1	1.000	4.25	4.25
ICF1	1.000	6.56	6.56
IFL1	1.000	10.28	10.27
NRR2	1.000	4.36	4.34
ICF2	1.000	6.85	6.85
IFL2	1.000	10.05	10.04
NRR3	1.000	4.04	4.04
ICF3	1.000	6.57	6.57
IFL3	1.000	9.38	9.34

NRR= Non-return rate
IFL= length of service period
ICF= interval from calving to first breeding

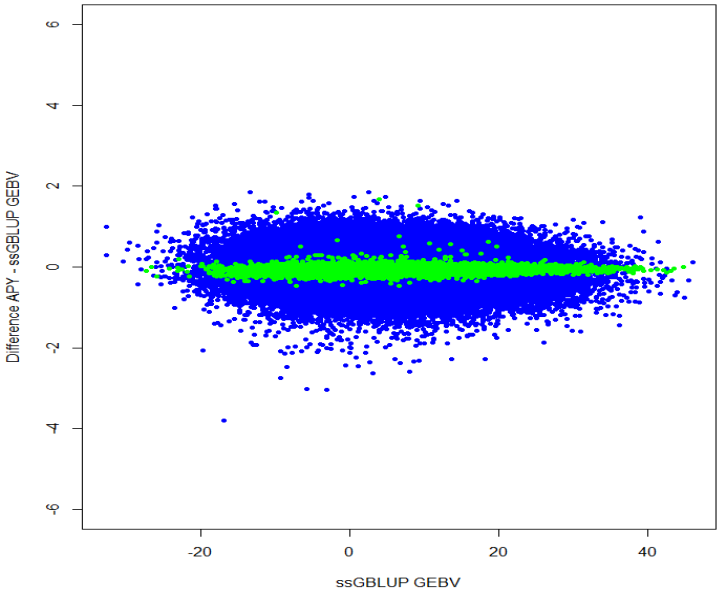
0= Heifer trait
1-3= cow trait 1-3 parity

Luke
NATURAL RESOURCES INSTITUTE FINLAND

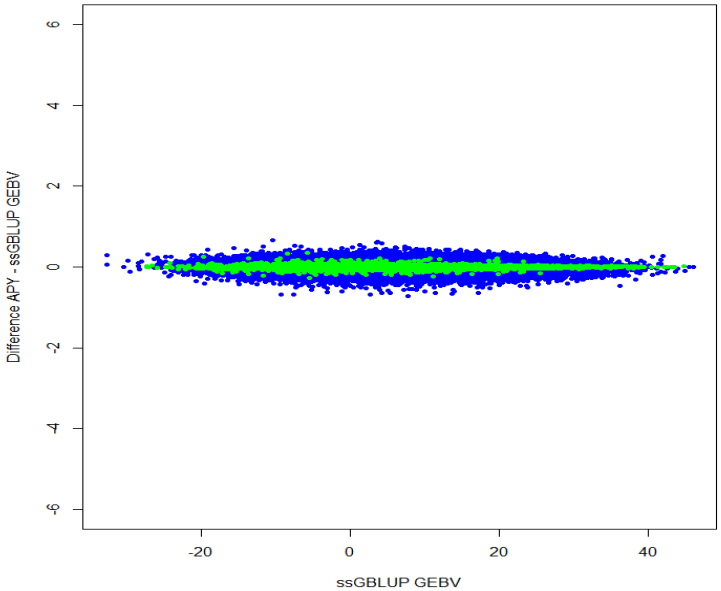
IFL2, 10,000 or 20,000 core

ssGBLUP std of GEBV (genotyped) = 10.05

10,000 in core



20,000 in core



Correlations:

Animals	10,000 in core	20,000 in core
Core	1.0000	1.0000
Young	0.9990	0.9999

Conclusions

- APY allowed reducing computing time substantially
  - The smaller the core the lower computing time

- Alternative computing strategies to $\mathbf{A}_{22}^{-1}\,\mathbf{d}$ gave good results:
  - CHOLMOD → Increased computing time minimally but memory need was increased and remained acceptable

➔ Single-step models with many genotyped animals can be analyzed
  - APY works logically: the more animals in core the better match

- Simply taking animals with highest number of progeny worked here

Thank you!

Thank you!

Luke
NATURAL RESOURCES
INSTITUTE FINLAND