# MiX99 single-step models & supporting programs

MiX99 course on genomic prediction

COURSE DAY 1, March 9th, 2026

# Contents

- Theory and motivation
- Model alternatives & their RAM requirements
- How to CLIM a standard ssGBLUP model
- The good, the bad and the ugly matrix formats
- Too many alternatives for $\mathbf{G}^{-1}$
- What, how, why or why not ssGTBLUP
- A view to ssSNPBLUP

Luke

# Theory and motivation

Consider a single trait (individual "animal") model

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Wu} + \mathbf{e},$$

where **y**= vector of observations,

**b** & **u**= vectors of fixed and random effects,

**X** & **W**= incidence matrices relating **b** & **u** to proper **y**.

Assume: **e** ~ N(**0**, **R**). Often $\mathbf{R}=\mathbf{I}\sigma_e^2$.

Assume: **u** ~ N(**0**, $\mathbf{V}_u\ \sigma_u^2$) where $\mathbf{V}_u$ is a relationship matrix.

Different relationship matrix based models:

- Animal model or PBLUP: $\mathbf{V}_u = \mathbf{A}$, the pedigree-based relationship matrix
- Genomic BLUP or GBLUP: $\mathbf{V}_u = \mathbf{G}$, the genomic relationship matrix
- single-step GBLUP or ssGBLUP: $\mathbf{V}_u = \mathbf{H}$, the unified relationship matrix of **A** and **G**

Different model names refer to the differences in the relationship matrix.

Mixed model equations:

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{W} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{W} + \sigma_u^{-2}\mathbf{V}_u^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix}$$

For ssGBLUP, the inverse of the relationship matrix is $\mathbf{V}_u^{-1} = \mathbf{H}^{-1}$ being

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

where

- $\mathbf{A}^{-1}$ is based on the pedigree relationships (sparse & easy by Henderson's rules),
- $(\mathbf{A}_{22})^{-1}$ is based on the pedigree relationships for the genotyped animals (can use parts of $\mathbf{A}^{-1}$),
- $\mathbf{G}^{-1}$ is based on genomic information (dense).

Typically: $\mathbf{G} = \mathbf{Z}_c\mathbf{B}\mathbf{Z}_c' + \mathbf{C}$ ,

where $\mathbf{Z}_c$ is centered marker matrix, **B** is scaling matrix and **C** is a regularization matrix

The dense nature of **G** and its inverse poses some computational challenges.

# Sparse vs. dense information



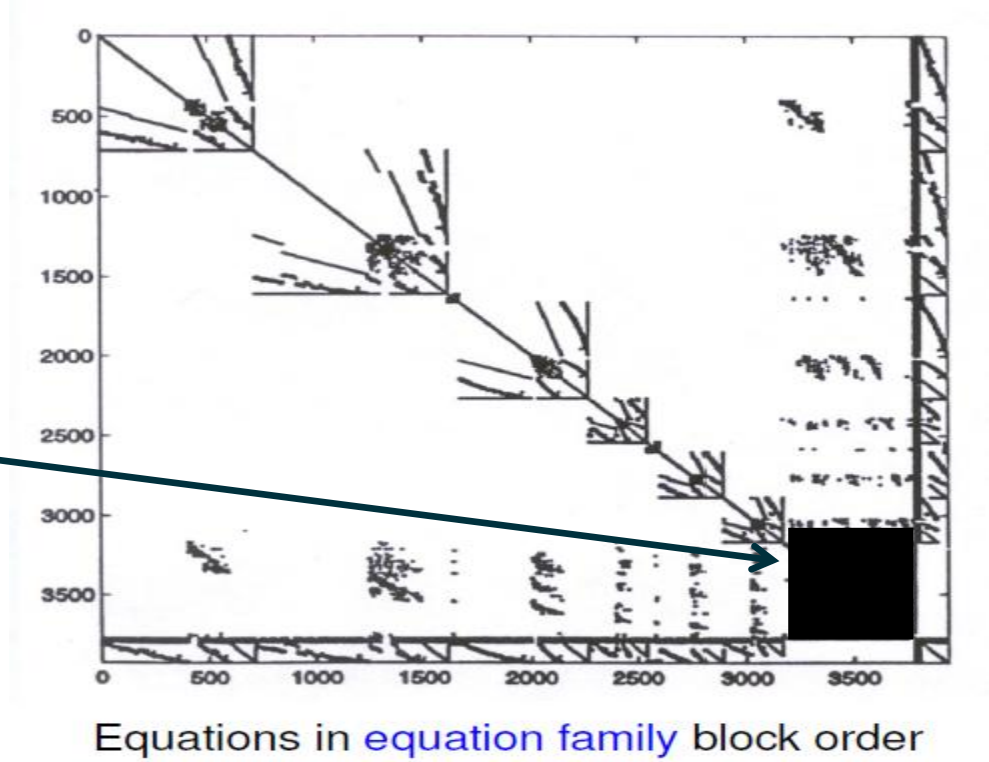
- Traditional evaluations have sparse information in animal model ($\mathbf{A}^{-1}$ is sparse)
  - Number of non-zero connections between different unknowns is low:

- Genomic information is dense
- The more dense matrix the more computations needed

The block $\mathbf{G}^{-1} - \mathbf{A}_{22}^{-1}$ increases computations **quadratically** in number of genotyped animals

although size of the genotype matrix $\mathbf{Z_c}$ increases **linearly** in number of genotyped animals.

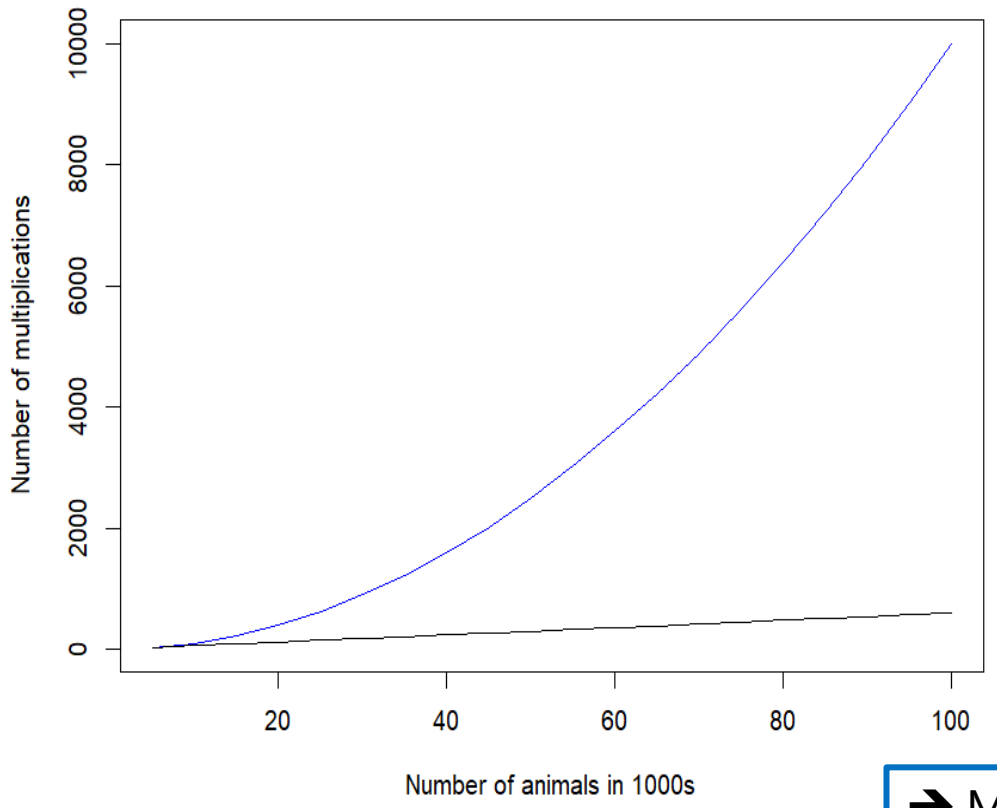
In practice, dense blocks are distributed everywhere in the matrix.




MiX99 does not make the coefficient matrix when estimating breeding values: uses so called iteration on data computations. Non-zero coefficients are computed when reading the data file.

The relationship matrix structures are in the pedigree for $\mathbf{A}^{-1}$ but as dense matrix for $\mathbf{G}^{-1}$.

# Number of computations by $\mathbf{A}^{-1}$ (Henderson's rules, black) and $\mathbf{G}^{-1}$ (dense square matrix, blue)




GBLUP and ssGBLUP scale quadratically by the number of genotyped animals.

Traditional animal model (PBLUP) scales linearly by the number of pedigree animals

The differences in RAM memory need is even larger because instead of making $\mathbf{A}^{-1}$, Henderson's rules need only the pedigree list (parents of every individual), but $\mathbf{G}^{-1}$ needs to be fully stored.

➔ Many alternative computational approaches for ssGBLUP have been proposed and some are available in MiX99.

# Model alternatives & their RAM requirements

# Many approaches for single-step



| | Precomputed | Pre-program | File |
|---|---|---|---|
| ssGBLUP | $\mathbf{G}^{-1}$ | hginv | Sparse ijvalue or lower dense of size *n* by *n* |
| APY | $\mathbf{G}^{-1}$ of APY | hginv | Sparse ijvalue or lower dense+sparse non-core |
| ssGTBLUP | **T** matrix | T48eig_make | Rectangular **T** of size *m* by *n* |
| C-ssGTBLUP | $\mathbf{Z}_c$ & $\mathbf{K}^{-1}$ | T48eig_make | Rectangular $\mathbf{Z}_c$; square dense $\mathbf{K}^{-1}$ of size *m* by *m* |
| Fully C-ssGTBLUP | $\mathbf{K}^{-1}$ | T48eig_make | Square dense $\mathbf{K}^{-1}$ of size *m* by *m* |
| ssSNPBLUP | - | - | - |

C- = Exact componentwise

ssGBLUP: uses standard single-step genomic BLUP approach, computations scale quadratically by the number of genotyped
APY: like ssGBLUP but uses an approximate $\mathbf{G}^{-1}$ matrix that is mostly sparse

ssGTBLUP: uses Woodbury matrix identity on $\mathbf{G}^{-1}$ such that the computations scale linearly by the number of genotyped
(Fully) C-ssGTBLUP: like ssGTBLUP but has a lower preprocessing need
ssSNPBLUP: an augmented version of ssGTBLUP with minimal prepreprocessing need and similar scalability as C-ssGTBLUP

Luke

# Single-step models: differences in MiX99



| | Precomputed | Pre-program | RAM need (bytes) | Marker solutions | Notes |
|---|---|---|---|---|---|
| ssGBLUP | $\mathbf{G}^{-1}$ | hginv | $8n^2$ | No | Small $n$ |
| APY | $\mathbf{G}^{-1}$ of APY | hginv | $8(n_c n+(n-n_c))$ | No | Approximate |
| ssGTBLUP | **T** matrix | T48eig_make | $8nm$ | No | Medium $n$ |
| C-ssGTBLUP | $\mathbf{Z}_c$ & $\mathbf{K}^{-1}$ | T48eig_make | $8(nm+m^2)$ | Yes | Testing purposes only |
| Fully C-ssGTBLUP | $\mathbf{K}^{-1}$ | T48eig_make | $8m^2+nm/5$ (packed) | Yes | Medium to large $n$ |
| ssSNPBLUP | - | - | $nm/5$ (packed) | Yes | Very large $n$ |

C- = Componentwise

$n$= number of genotyped
$n_c$ = number of core individuals
$m$= number SNP markers

ssGTBLUP can use approximate approach by eigendecomposition of the **T** matrix. Useful for small n, large m cases.

Chromosomewise approximate approach by Ødegård et al. (2018) is also available but has been studied very little.

Many alternatives available. Is there a clear rule on which is the best?

Luke

# Single-step models: RAM use examples with m=50000 markers



| | RAM need (bytes) | N=50,000 (GB) | N=300,000 (GB) | N=1 million (GB) | N=5 million (GB) |
|---|---|---|---|---|---|
| ssGBLUP | $8n^2$ | 20 | 720 | 8 TB | 200 TB |
| APY ($n_c$=15000) | $8(n_c n+(n-n_c))$ | 6 | 36 | 120 | 600 |
| ssGTBLUP | $8nm$ | 20 | 120 | 400 | 2 TB |
| C-ssGTBLUP | $8(nm+m^2)$ | 40 | 140 | 420 | 2 TB |
| Fully C-ssGTBLUP | $8m^2+nm$ | 22 | 35 | 70 | 270 |
| - Packed | $8m^2+nm/5$ | 21 | 23 | 30 | 70 |
| ssSNPBLUP | $nm$ | 3 | 15 | 50 | 250 |
| - Packed | $nm/5$ | 1 | 3 | 10 | 50 |

C- = Componentwise

$n$= number of genotyped
$n_c$ = number of core individuals
$m$= number SNP markers

Other factors to consider: preprocessor computing time and solver convergence

Luke

# How to CLIM a standard ssGBLUP model

# CLIM of a basic pedigree-based animal model



Data file information

```
DATAFILE   ../data/9_SNP_WT_groups.dat
MISSING    -9
INTEGER    row ones ID
REAL       wt y trueDGV wght g1 g2 g3 g4
```

Pedigree file information

```
PEDFILE     ../data/sim_ped_mod.ped
PEDIGREE    ID am

INBRFILE    ../data/sim_ped_mod.inbr
INBREEDING PEDIGREECODE=1 FINBR=3
```

Variance components

```
PARFILE    ../data/AM.var
TMPDIR     ./tmp
```

Fixed and random effects of model

```
MODEL
  y = g1 g2 g3 g4 ones ID ! WEIGHT=wght
```

# CLIM of a basic ssGBLUP



- Single-step GBLUP requires inverse of the genomic relationship matrix (GRM) in addition to the pedigree information:

- 2 alternative ways to give:
  - **ssGBLUP inverse GRM file**
  - iGFILE + iA22PEDIGREE

```
DATAFILE   ../data/9_SNP_WT_groups.dat
MISSING    -9
INTEGER    row ones ID
REAL       wt y trueDGV wght g1 g2 g3 g4

ssGBLUP    LOWER ../data/iGL_w20.dat
# iGFILE     LOWER ../data/iGL_w20.dat
# iA22FILE   PEDIGREE

PEDFILE    ../data/sim_ped_mod.ped
PEDIGREE   ID am

INBRFILE   ../data/sim_ped_mod.inbr
INBREEDING PEDIGREECODE=1 FINBR=3

PARFILE    ../data/AM.var
TMPDIR     ./tmp

MODEL
  y = g1 g2 g3 g4 ones ID ! WEIGHT=wght
```

# The good, the bad and the ugly matrix formats

# $G^{-1}$ file formats

## ijval-format (Coordinate)

- Default in SSGBLUP
- Non-zero values
- Upper or lower triangle
- Symmetric matrix

$$\mathbf{G}^{-1} = \begin{bmatrix} 57 & -15 & 15 & 75 & 24 & -39 \\ -15 & 6 & -3 & -18 & -6 & 9 \\ 15 & -3 & 6 & 21 & 6 & -12 \\ 75 & -18 & 21 & 105 & 33 & -57 \\ 24 & -6 & 6 & 33 & 12 & -18 \\ -39 & 9 & -12 & -57 & -18 & 33 \end{bmatrix}$$

| i | j | value |
|---|---|---|
| 1 | 1 | 57 |
| 2 | 1 | -15 |
| 2 | 2 | 6 |
| 3 | 1 | 15 |
| 3 | 2 | -3 |
| 3 | 3 | 6 |
| 4 | 1 | 75 |
| 4 | 2 | -18 |
| 4 | 3 | 21 |
| 4 | 4 | 105 |
| 5 | 1 | 24 |
| 5 | 2 | -6 |
| 5 | 3 | 6 |
| 5 | 4 | 33 |
| 5 | 5 | 12 |
| 6 | 1 | -39 |
| 6 | 2 | 9 |
| 6 | 3 | -12 |
| 6 | 4 | -57 |
| 6 | 5 | -18 |
| 6 | 6 | 33 |

## Lower triangle dense format

- Lower triangle or symmetric matrix
- All values
- Has to be specified

```
6 0
1 2 3 4 5 6
 57
-15   6
 15  -3   6
 75 -18  21 105
 24  -6   6  33  12
-39   9 -12 -57 -18 33
```

File suffix rules:

| | |
|---|---|
| .bin | Fortran unformatted binary file |
| .raw | Stream binary file |

All others are assumed to be text files.

# Examples

- SSGBLUP ijval_file.dat ijval format text
- SSGBLUP ijval_file.bin Unformatted ijval binary
- SSGBLUP ijval_file.raw Stream binary ijval format

- SSGBLUP LOWER my_matrix.dat Lower triangle dense format text
- SSGBLUP LOWER my_matrix.bin Unformatted lower triangle dense binary
- SSGBLUP LOWER my_matrix.raw Stream binary lower triangle dense

Why lower triangle dense: often smaller file, faster to write and read

Why bin or raw format: takes less space, faster to read/write, exact values

Luke

# Marker file formats for single-step



Marker matrix data is required by
- mix99i
- Preprocessing programs like hginv and T48eig_make
- Post-analysis programs like predict_GEBV

- Marker files have
  - ID codes in the first column
  - Marker data coded as 0,1,2 for the number of counted alleles

- File formats:
  - Space separated marker data (default)
  - Fortran format read marker data
  - No space between markers (after ID code)

- Marker data stored in memory:
  - Double precision full (ZCFILE command)
  - One byte integers (default for SNPMATRIX command)
  - 5 markers packed to one byte integer (SNPMATRIX: USE=PACK)

Luke

# Too many alternatives for $G^{-1}$

Genomic relationship matrix has often form:

$$\mathbf{G} = \mathbf{Z_c B Z_c'} + \mathbf{C}\,,$$

where $\mathbf{Z}_c$ is centered marker matrix, **B** is scaling matrix and **C** is a regularization matrix.

- VanRaden method 1: $\mathbf{G} = \mathbf{Z_c Z_c'}/k$,

where $k = 2\sum_{i=1}^{m} p_i(1-p_i)$ and $p_i$ = allele frequency of marker $i$.

- VanRaden method 2: $\mathbf{G} = \mathbf{Z_c D Z_c'}$,

where the element $i$ of the diagonal matrix **D** is $\mathbf{D}_{ii} = 1/\big(2p_i(1-p_i)\big)$.

Possibility for no inverse!
- $m < n$
- Allele frequencies from data

- Regularization e: $\mathbf{C} = \mathbf{I}c$ where $c$ is a small number like 0.001.
- Regularization A: $\mathbf{G} = (1-w)\mathbf{Z_c B Z_c'} + w\mathbf{A}_{22}$ where $w$ is the residual polygenic proportion.

Many ways to make a genomic relationship matrix

# Making $G^{-1}$ using hginv



Common:

```
-m method : genomic matrix, method after -m is
    raw : use genotype data as such
    101 : 101 coding (-1,0,1), assumes genotype data has 012 coding
 center : center coding, i.e. PvR1 without scaling by 2*sum(p*q)
    avg : center around average allele frequency.
   PvR1 : P.VanRaden method 1, singular if data allele frequencies
  PvR1m : PvR1 with diag(G) multiplied by  1.0010
  PvR1a : PvR1 with diag(G) increased by 0.10000E-02
  PvR1am: PvR1 with G*(1-b)+I*b with b= 0.10000E-02
   PvR2 : P.VanRaden method 2 (-c m), singular if data allele frequencies
  PvR2m : PvR2 with diag(G) multiplied by  1.0010
   poly : blending G(PvR1) and A22: G_w = (1-w)*G+w*A22, see option -w
    edm : Euclidean distance norm matrix, see option -theta.
    ost : matrix B (see -ss) for single-step, same as giving -m poly -c dA -ss
   ost2 : like ost but use PvR2 instead of PvR1, or -m PvR2 -c dA -ss
   mean : centering by mean of marker values by SD.


 -c kval: scaling for G=ZDZ'/kval where kval is
    2pq : divide by 2*sum(p*q), default for PvR, PvR1, PvR1a, PvR1m, PvR1am
      m : divide by m=number of markers, default for PvR2 & PvR2m.
     m2 : divide by m/2.
     dA : multiply by trace(A22)/trace(ZDZ'), default for ost and poly.
    one : average diagonal of ZDZ' will be one (Forni et. al. GSE 2011)
    avg : divide by 2*m*avg_p*(1-avg_p), avg_p= average MAF.
    lsA : least squares c in (diag(A22) - diag(ZDZ')/c)^2.
     no : no scaling, default for 101 and raw.
```

# Hginv examples



- PvR1 with 0.001 added regularization giving text file (iGL_e.dat) in lower dense format:

hginv_para -lower -a base_af.dat -m PvR1a mygeno.dat iGL_e.dat

- PvR1 blending genomic with 10% **A** matrix giving an unformatted binary file (iGL_w20.bin) in lower dense format:

hginv_para -lower -w 0.10 -Alower A22.Lamat -a base_af.dat -m PvR1 -c 2pq mygeno.dat iGL_w20.bin

- Option –a uses the allele frequencies in the given file for centering marker data and in the scaling constant.
- SNP marker data file is assumed to be space-separated and the first column has the ID code.
  - Options to read marker data without spaces exist (-nospace) and using a Fortran format (-FMT)
- Many options exist for scaling, choosing markers etc. Later some more options covered.

Hint: use the –info option to see how the hginv has understood the options before running it.

# What, how, why or why not ssGTBLUP

## ssGTABLUP

Mäntysaari, E. A., R. D. Evans, and I. Strandén. 2017. Efficient single-step genomic evaluation for a multibreed beef cattle population having many genotyped animals. J. Anim. Sci. 95:4728–4737. https://doi.org/10.2527/jas2017.1912.



Assumes the **G** matrix has form: $\mathbf{G} = \mathbf{Z}_c\mathbf{B}\mathbf{Z}_c' + w\mathbf{A}_{22}$

where $\mathbf{Z}_c$ is centered marker matrix, $w$ is the residual polygenic proportion and the $\mathbf{A}_{22}$ matrix is pedigree-based relationship matrix of genotyped animals.

In VanRaden 1, the scaling matrix **B** is $\mathbf{B} = \mathbf{I}\frac{1-w}{k}$ with the scaling constant $k = 2\sum_{i=1}^{m} p_i(1-p_i)$.

## ssGTeBLUP

Assumes the **G** matrix has form: $\mathbf{G} = \mathbf{Z}_c\mathbf{B}\mathbf{Z}_c' + \varepsilon\mathbf{I}$

where $\mathbf{Z}_c$ is centered marker matrix, $\varepsilon$ is a small number like 0.001.

In VanRaden 1, the scaling matrix **B** is $\mathbf{B} = \mathbf{I}\frac{1}{k}$

# Traditional ssGTABLUP (ssGTeBLUP is similar, not shown)



Assumes the **G** matrix has form: $\mathbf{G} = \mathbf{Z}_c\mathbf{B}\mathbf{Z}_c' + w\mathbf{A}_{22}$

where $\mathbf{Z}_c$ is centered marker matrix, $w$ is the residual polygenic proportion and the $\mathbf{A}_{22}$ matrix is pedigree-based relationship matrix of genotyped animals.

(Woodbury matrix identity):

$$\mathbf{G}^{-1} = \frac{1}{w}\mathbf{A}_{22}^{-1} - \frac{1}{w}\mathbf{A}_{22}^{-1}\mathbf{Z}_c\left(\frac{1}{w}\mathbf{Z}_c'\mathbf{A}_{22}^{-1}\mathbf{Z}_c + \mathbf{B}^{-1}\right)^{-1}\mathbf{Z}_c'\mathbf{A}_{22}^{-1}\frac{1}{w}$$

$$= \frac{1}{w}\mathbf{A}_{22}^{-1} - \mathbf{T}'\mathbf{T}$$

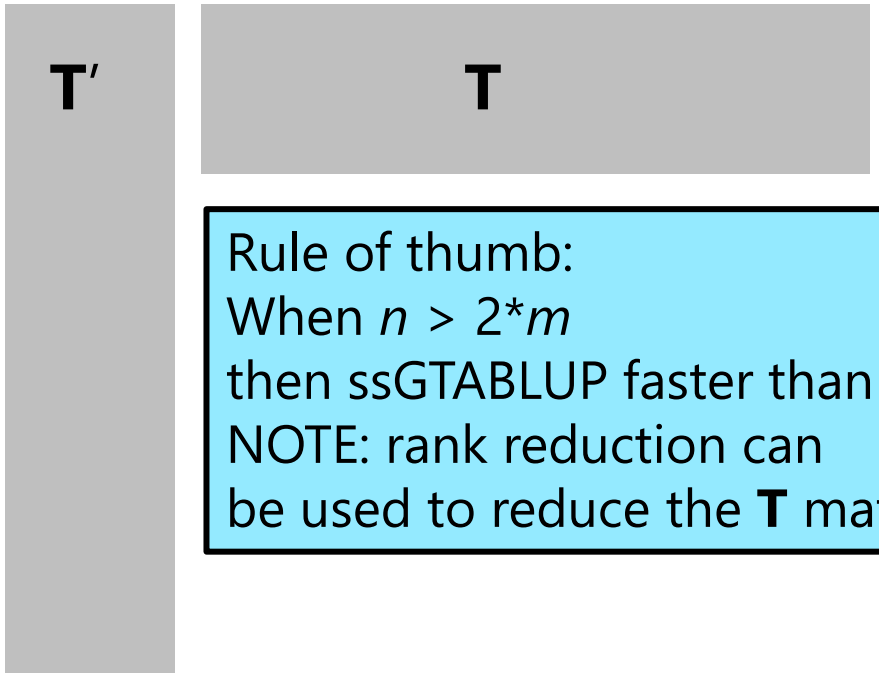
where $\mathbf{T} = \left(\frac{1}{w}\mathbf{Z}_c'\mathbf{A}_{22}^{-1}\mathbf{Z}_c + \mathbf{B}^{-1}\right)^{-0.5}\mathbf{Z}_c'\mathbf{A}_{22}^{-1}\frac{1}{w}$

Matrix $\mathbf{T}$ has size *m* by *n*:

*m* = number of SNP markers

*n* = number of genotyped

- **T** can be computed by T48eig_make




Rule of thumb:
When *n* > 2**m*
then ssGTABLUP faster than ssGBLUP
NOTE: rank reduction can
be used to reduce the **T** matrix size.

# Component-wise ssGTABLUP

Mäntysaari, E. A., M. Koivula, and I. Strandén. 2020. Symposium review: Single-step genomic evaluations in dairy cattle. J. Dairy Sci. 103:5314–5326. https://doi.org/10.3168/jds.2019-17754.



There are two ways for ssGTABLUP in MiX99:

- Traditional: make the **T** matrix and use it
- Component-wise: based on the idea that PCG makes the computations from right to left:

$$\mathbf{G}^{-1} = \frac{1}{w}\mathbf{A}_{22}^{-1}\mathbf{s} - \frac{1}{w}\mathbf{A}_{22}^{-1}\mathbf{Z}_c\left(\frac{1}{w}\mathbf{Z}_c'\mathbf{A}_{22}^{-1}\mathbf{Z}_c + \mathbf{B}^{-1}\right)^{-1}\mathbf{Z}_c'\mathbf{A}_{22}^{-1}\frac{1}{w}\mathbf{s}$$

$$= \frac{1}{w}\mathbf{A}_{22}^{-1}\mathbf{s} - \frac{1}{w}\mathbf{A}_{22}^{-1}\mathbf{Z}_c\mathbf{K}^{-1}\mathbf{Z}_c'\mathbf{A}_{22}^{-1}\frac{1}{w}\mathbf{s}$$

where $\mathbf{K}^{-1} = \left(\frac{1}{w}\mathbf{Z}_c'\mathbf{A}_{22}^{-1}\mathbf{Z}_c + \mathbf{B}^{-1}\right)^{-1}$ has been precomputed. Solver computations are efficient.

MiX99 has *2 component-wise* approaches:

- Explicit: uses the centered marker matrix $\mathbf{Z}_c$ and $\mathbf{K}^{-1}$ (NOT recommended to use!)
- Fully: uses $\mathbf{K}^{-1}$ and the original non-centered marker matrix, $\mathbf{Z}_c$ is built on-the-fly from the marker data!

Component-wise ssGTABLUP allows computing SNP marker solutions.

# Example: ssGTABLUP

Traditional ssGTABLUP

```
...
TAFILE     TA.bin
iA22FILE  PEDIGREE
# Alternatively: SSGTABLUP TA.bin

PEDFILE        pruned.ped
PEDIGREE       G am
INBRFILE       pruned.inbr
INBREEDING PEDIGREECODE=1 FINBR=3
...
MODEL
...
```

Explicit componentwise ssGTABLUP

```
...
ZCFILE    Zc.bin
ICFILE    iC.bin
iA22FILE PEDIGREE

PEDFILE        pruned.ped
PEDIGREE       G am
INBRFILE       pruned.inbr
INBREEDING PEDIGREECODE=1 FINBR=3
...
MODEL
...
```

# Options for T48eig_make

```
-info       : print these comments, current option values and stop.
-nthr n     : number of threads used in parallel computing is set to n.
-sread      : use single thread file reading.
-pread      : use parallel file reading (default).
-m method   : marker matrix centering method.
    raw     : no scaling, i.e., use genotype data as such.
    101     : 101 coding (-1,0,1), assumes genotype data has 012 coding.
     AF     : center by allele frequencies (see -a).
-a a_file   : allele frequency file for -m AF and -c 2pq (see -m & -c),
              File has format: <allele frequency>
              First line is for the first marker, 2nd for 2nd marker ...
-c kval     : scaling for G=ZDZ'/kval where kval is
        2pq : divide by 2*sum(p_i*(1-p_i)), p_i allele frequency of marker i.
          m : divide by m=number of markers
         m2 : divide by m/2.
         dA : multiply by trace(A22)/trace(G), see -scale
        one : average diagonal of G will be one (Forni et. al. GSE 2011)
        avg : divide by 2*m*avg_p*(1-avg_p), avg_p= average allele freq.
         no : no scaling.
```

```
ssGTeBLUP   : specic options:
-step N     : reduce memory allocation in -eig and -diG options. The computations are
              in multiples of N *n_genotyped. Default N is number of genotyped.
ssGTABLUP   : calculation method for inv(A22) (only one can be used):
       -IOP : use iteration on pedigree in the calculations.
        -IM : use iteration in memory in the calculations.
       -CHM : use Cholmod approach in the calculations.
       -PAR : use MKL PARDISO with OpenMP disection in the calculations (default).
  -metisPAR : in -PAR use METIS in disection (use this if -PAR fails).
```

```
ssGTABLUP   : memory use
   -memhigh : large memory use.
   -memmed  : medium memory use (default).
   -memlow  : low memory use by byte packed marker matrix.
   -memlows : like -memlow but also use single precision inv(C).
   -nblk <v>: computing buffer size.
```

```
-FMT FMT    : optional format to read genotyped file, eg. -FMT "(i9,1x,60000i1)".
-nospace    : no space between the marker genotypes but at least 1 space after the id code (1st column).
-first c    : change default column number (2) of the first marker to be c.
-add  e     : ssGTeBLUP: assign the added value e.
-ignore     : do not require 012 allele coding (used in mthd 'raw').
-eig  v     : ssGTeBLUP: use eigenvalue decomposition.
              The value v is proportion of eigenvalues explaining variation,
              where v is number in the interval (0,1].
-rpg  w     : ssGTABLUP: assign RPG weight of A22 to be w.
-P ped_file : ssGTABLUP: pedigree file (input)
-F F_file   : ssGTABLUP: inbreeding coefficients file (input).
-Fcol c     : ssGTABLUP: change default inbreeding coefficient column (3) in F_file.
-iGm <file> : ssGTABLUP: optional inverse metafounder (MF) matrix file. Format: <MF 1> <MF 2> <value>
-groups n   : ssGTABLUP(old): include unknown parent groups (negative parent), n=maximum number of groups.
-ZC iCfile  : new stepwise ssGTABLUP: compute and write inv(C) (output).
            : inv(C)= inv(Z'inv(A22)Z/w + (k/(1-w))I ), k=scaling kval.
              Centered Z=(M-2P) matrix file is written to fileout. Default: TA matrix in fileout.
```

Luke

# T48eig_make and component-wise ssGTABLUP



| | | |
|---|---|---|
| `T48eig_make` | | |
| `-nblk 1000` | : | 1000 blocks in computations |
| `-nthr 10` | : | 10 CPU threads |
| `-c dA` | : | scaling by tr($\mathbf{A}_{22}$)/tr($\mathbf{G}$) |
| `-rpg 0.2` | : | residual polygenic proportion $w$=0.2 |
| `-a base_af.dat` | : | allele frequencies for centering |
| `-FMT "(i10,26x,50240i1)"` | : | input format of genotypes |
| `-P mypedigree.ped` | : | pedigree file for $\mathbf{A}_{22}$ computations |
| `-F myinbreeding.inbr -Fcol 3` | : | inbreeding coefficients (column 3) |
| **`-ZC iC.bin`** | : | `Output:` $\mathbf{K}^{-1}$ matrix for ICFILE |
| `mygenotypes.dat` | : | `Input:` genotype file |
| **`Zc.bin`** | : | `Output:` centered marker matrix $\mathbf{Z}_c$ for ZCFILE |

```
T48eig_make -nblk 1000 -nthr 10 -c dA -rpg 0.2 -a base_af.dat -FMT "(i10,26x,50240i1)" \
            -P mypedigree.ped -F myinbreeding.inbr -Fcol 3 \
            -ZC iC.bin mygenotypes.dat Zc.bin
```

When **no** –ZC option is given, the result is a **T** matrix for traditional ssGTABLUP.

# Different component-wise ssGTABLUP:s



Explicit component-wise ssGTABLUP:

```
ZcFile      ../geno_data_nocand/TA_w20_TZ_ref.bin
iCfile      ../geno_data_nocand/iC_w20_TZ_ref.bin
iA22File    PEDIGREE
```

Fully component-wise ssGTABLUP with int-1 packed SNP-matrix:

```
SNPMATRIX  FIRST=2 LAST=50241 CENTER=p FORMAT='(i10,26x,50240i1)'
SNPFILE    ../../geno_data_nocand/ICBF_2018_10_genotypes_in_ped_ref.dat
CENTERFILE base_af_2col.dat
iCfile     ../../geno_data_nocand/iC_w20_TZ_ref_med_OMP.raw
iA22File   PEDIGREE
```

No ZcFILE

T48eig_make note: There is no need to make Zc file for the fully component-wise ssGTABLUP
→ The output file for this file is then left to be a minus sign.

```
T48eig_make -nblk 1000 -nthr 10 -c dA -rpg 0.2 -a base_af.dat -FMT "(i10,26x,50240i1)" -P mypedigree.ped -F myinbreeding.inbr -Fcol 3 ...
            -ZC iC.bin mygenotypes.dat -
```

Almost 1M genotyped & 50,000 SNP markers:
Explicit component-wise: ~390GB RAM
Fully component-wise: ~76GB RAM.

Luke

# A view to ssSNPBLUP

# ssSNPBLUP is an augmented version of ssGTABLUP



MME are

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{W} & \mathbf{0} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{W} + \sigma_u^{-2}\mathbf{H}_C^{-1} & -\sigma_u^{-2}\mathbf{K}_C \\ \mathbf{0} & -\sigma_u^{-2}\mathbf{K}_C' & \sigma_u^{-2}\mathbf{K} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \\ \hat{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{0} \end{bmatrix}$$

where

$\mathbf{H}_C^{-1} = \mathbf{A}^{-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{-1} - \mathbf{A}_{22}^{-1} \end{bmatrix}$, $\mathbf{K}_C = \begin{bmatrix} \mathbf{0} \\ \mathbf{C}^{-1}\mathbf{Z}_c \end{bmatrix}$ matrix is from the marker effects to genotypes,

$\mathbf{K} = \mathbf{Z}_c'\mathbf{C}^{-1}\mathbf{Z}_c + \mathbf{B}^{-1}$, and $\mathbf{C} = w\mathbf{A}_{22}$.

Note: 1) No $\mathbf{K}$ matrix is explicitly formed by MiX99

2) Genomic information is only in $\mathbf{K}_C$ and $\mathbf{K}$.

3) Absorbing the marker effect lines gives ssGTABLUP.

Liu Z, Goddard ME, Reinhardt F, Reents R. A single-step genomic model with direct estimation of marker effects. J Dairy Sci. 2014;97:5833–50.

Vandenplas, J., ten Napel, J., Darbaghshahi, S. N., Evans, R., Calus, M. P., Veerkamp, R., et al. (2023). Efficient large-scale single-step evaluations and indirect genomic prediction of genotyped selection candidates. *Genet. Sel. Evol.* 55, 1–17. doi:10.1186/s12711-023-00808-z

## ssGTABLUP

```
DATAFILE   ../data/9_SNP_WT_groups_2traits.dat
MISSING    -9
INTEGER    row ones ID
REAL       wt y y2 trueDGV wght g1 g2 g3 g4
DATASORT   PEDIGREECODE=ID

SNPMATRIX  FIRST=2 LAST=1001 CENTER=p FORMAT='(i2,1x,1000i1)'
SNPFILE    ../data/9_Z0_id_16last_nospace.dat
CENTERFILE ../data/base_af_1000.dat
iCFILE     ../data/iC_w20_OMP.raw
iA22FILE   PEDIGREE

PEDFILE    ../data/sim_ped_mod.ped
PEDIGREE   ID am

INBRFILE   ../data/sim_ped_mod.inbr
INBREEDING PEDIGREECODE=1 FINBR=3

PARFILE    ../data/AM_2tr.var
TMPDIR     ./tmp

MODEL
  y  = ones ID ! WEIGHT=wght
  y2 = ones ID ! WEIGHT=wght
```

## ssSNPBLUP – no preprocessing

```
DATAFILE   ../data/9_SNP_WT_groups_2traits.dat
MISSING    -9
INTEGER    row ones ID
REAL       wt y y2 trueDGV wght g1 g2 g3 g4
DATASORT   PEDIGREECODE=ID

SNPMATRIX  FIRST=2 LAST=1001 FORMAT='(i2,1x,1000i1)' CENTER=p SCALE=p
SNPFILE    ../data/9_Z0_id_16last_nospace.dat
CENTERFILE ../data/base_af_1000.dat
SSSNPBLUP  GTA 0.20
iA22FILE   PEDIGREE

PEDFILE    ../data/sim_ped_mod.ped
PEDIGREE   ID am

INBRFILE   ../data/sim_ped_mod.inbr
INBREEDING PEDIGREECODE=1 FINBR=3

PARFILE    ../data/AM_2tr.var
TMPDIR     ./tmp

MODEL
  y  = ones ID ! WEIGHT=wght
  y2 = ones ID ! WEIGHT=wght
```

```
T48eig_make -nospace -c 2pq -rpg 0.2 -a af.dat -P sim_ped_mod.ped -F sim_ped_mod.inbr -Fcol 3 -ZC iC_w20_OMP.raw 9_Z0_id_16last_nospace.dat -
```

An alternative way to give ssSNPBLUP in MiX99 exists: shown later

# A short summary

# Too much RAM about nothing or can we solve single step?

| | RAM need (bytes) | N=50,000 (GB) | N=300,000 (GB) | N=1 million (GB) | N=5 million (GB) |
|---|---|---|---|---|---|
| ssGBLUP | $8n^2$ | 20 | 720 | 8 TB | 200 TB |
| APY ($n_c$=15000) | $8(n_c n+(n-n_c))$ | 6 | 36 | 120 | 600 |
| ssGTBLUP | 8nm | 20 | 120 | 400 | 2 TB |
| C-ssGTBLUP | $8(nm+m^2)$ | 40 | 140 | 420 | 2 TB |
| Fully C-ssGTBLUP | $8m^2+nm$ | 22 | 35 | 70 | 270 |
| - Packed | $8m^2+nm/5$ | 21 | 23 | 30 | 70 |
| ssSNPBLUP | nm | 3 | 15 | 50 | 250 |
| - Packed | nm/5 | 1 | 3 | 10 | 50 |

# CLIM commands for single-step



- ssGBLUP: SSGBLUP [LOWER] my_inverse_G_matrix
- ssGTBLUP: SSGTBLUP [TAFILE|TEFILE] my_T_matrix
- Fully component-wise ssGTBLUP (gives also marker solutions)

  SNPMATRIX [USE=BYTE|PACK] FIRST= LAST= [CENTER= FORMAT=]

  SNPFILE my_geno.dat

  [CENTERFILE my_CENTER.dat]

  iCFILE my_iC_file

  iA22FILE PEDIGREE
- Alternative way to give ssSNPBLUP (gives also marker solutions)

  SSSNPBLUP GTA|GTe value my_geno.dat

  SNPMATRIX [USE=BYTE|PACK] FIRST= LAST= [CENTER= FORMAT= SCALE=]

  [CENTERFILE my_CENTER.dat]

Next: unknown parent groups, preconditioner, convergence, solving times…

Luke