Solving large single-step models with MiX99

Ismo Strandén

MiX99 course: test-day models and single step genomic prediction

COURSE DAY 2, April 11th, 2025

Contents

Single-step BLUP (ssGBLUP) allows simultaneously combining genomic information with traditional pedigree information.

Model: $\mathbf{y} = \mathbf{Xb} + \mathbf{Wa} + \mathbf{e}$ where $\mathbf{e} \sim (\mathbf{0}, \mathbf{R}\sigma_e^2)$ and $\mathbf{a} \sim (\mathbf{0}, \mathbf{H}\sigma_a^2)$

Mixed model equations:

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{W} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{W} + \sigma_a^{-2}\mathbf{H}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix}$$

where the inverse of the relationship matrix is

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

where

- $\mathbf{A}^{-1}$ is based on the pedigree relationships (sparse & easy to compute),
- $(\mathbf{A}_{22})^{-1}$ is based on the pedigree relationships for the genotyped animals,
- $\mathbf{G}^{-1}$ is based on genomic information $\mathbf{Z}_c$ (dense).

Typically: $\mathbf{G} = \mathbf{Z}_c \mathbf{B} \mathbf{Z}_c' + \mathbf{C}$ where $\mathbf{Z}_c$ is centered marker matrix, $\mathbf{B}$ is scaling matrix and $\mathbf{C}$ is a regularization matrix

Sparse vs. dense information

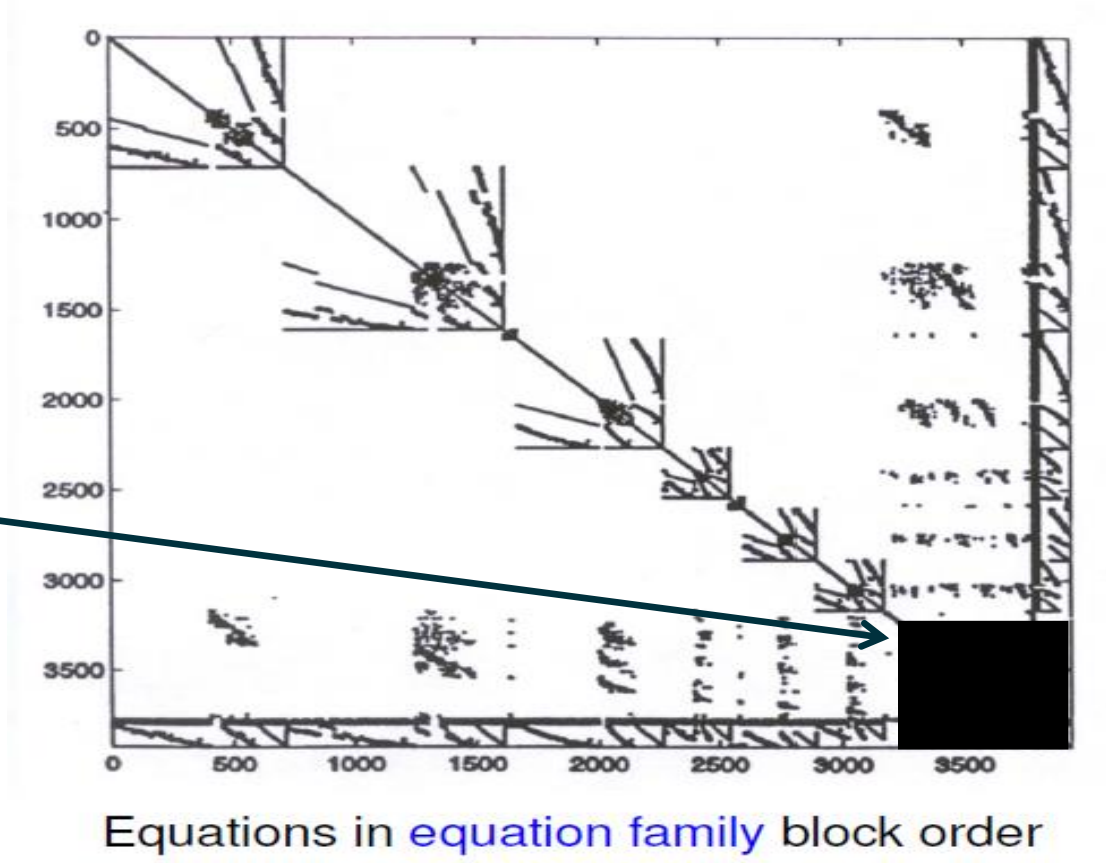
- Traditional evaluations have sparse information in animal model
  - Number of non-zero connections between different unknowns is low:

- 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 throughout the matrix.



Many approaches for single-step

	Precomputed	Pre-program	Files
ssGBLUP	$\mathbf{G}^{-1}$	hginv	Sparse ijvalue, Lower dense
APY	$\mathbf{G}^{-1}$ of APY	hginv	Sparse ijvalue, Lower dense+sparse non-core
ssGTBLUP	**T** matrix	T48eig_make	Rectangular
C-ssGTBLUP	$\mathbf{Z}_c$ & $\mathbf{K}^{-1}$	T48eig_make	Rectangular $\mathbf{Z}_c$; square dense $\mathbf{K}^{-1}$
Fully C-ssGTBLUP	$\mathbf{K}^{-1}$	T48eig_make	square dense $\mathbf{K}^{-1}$
ssSNPBLUP	-	-	-

C- = Componentwise

File suffix rules:

.bin Fortran unformatted binary file

.raw Fortran stream binary file

All others are assumed to be text files.

Only the lower dense format has to be specified.
All other file formats are assumed.

Example:

ssGBLUP LOWER iGlower.dat

CLIM of a basic ssGBLUP

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

- 2 alternatives:

- 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
```

Regular ssGBLUP model with genetic groups: which can give the same breeding values?


```
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

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

```
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_QP.dat
# iGFILE     LOWER data/iGL_w20_QP.dat
# iA22FILE   PEDIGREE

PEDFILE    data/sim_ped_mod.ped
PEDIGREE   ID am+p

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

PARFILE    data/AM.var

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

```
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+p

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

PARFILE    data/AM.var

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

And which is correct?

Genetic groups: $\mathbf{y = Xb + WQv + Wa + e}$ where $\mathbf{v} \sim (\mathbf{0}, \gamma^{-1}\mathbf{I}\sigma_a^2)$ with γ scaling factor.

γ scaling factor is often from 0.3 to 1.0

MME for the ssGBLUP model with genetic groups are

$$\begin{bmatrix} \mathbf{X'R^{-1}X} & \mathbf{X'R^{-1}WQ} & \mathbf{X'R^{-1}W} \\ \mathbf{Q'W'R^{-1}X} & \mathbf{Q'W'R^{-1}WQ} + \gamma\mathbf{I}\sigma_a^{-2} & \mathbf{Q'W'R^{-1}W} \\ \mathbf{W'R^{-1}X} & \mathbf{W'R^{-1}WQ} & \mathbf{W'R^{-1}W} + \mathbf{H^{-1}}\sigma_a^{-2} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{v}} \\ \hat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'R^{-1}y} \\ \mathbf{Q'W'R^{-1}y} \\ \mathbf{W'R^{-1}y} \end{bmatrix}$$

where $\mathbf{H^{-1}} = \mathbf{A^{-1}} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G^{-1}} - \mathbf{A_{22}^{-1}} \end{bmatrix}$.

Estimates of the breeding values are $\hat{\mathbf{a}}_d = \mathbf{Q}\hat{\mathbf{v}} + \hat{\mathbf{a}}$

After QP transformation, the MME are

$$\begin{bmatrix} \mathbf{X'R^{-1}X} & \mathbf{0} & \mathbf{X'R^{-1}W} \\ \mathbf{0} & \mathbf{Q'H^{-1}Q}\sigma_a^{-2} + \gamma\mathbf{I}\sigma_a^{-2} & -\mathbf{Q'H^{-1}}\sigma_a^{-2} \\ \mathbf{W'R^{-1}X} & -\mathbf{H^{-1}Q}\sigma_a^{-2} & \mathbf{W'R^{-1}W} + \mathbf{H^{-1}}\sigma_a^{-2} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{v}} \\ \hat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'R^{-1}y} \\ \mathbf{0} \\ \mathbf{W'R^{-1}y} \end{bmatrix}$$

Alternatively: $\mathbf{y} = \mathbf{Xb} + \mathbf{W}(\mathbf{Q} + \mathbf{J})\mathbf{v}_{\mathrm{QJ}} + \mathbf{W}\mathbf{a}_{\mathrm{QJ}} + \mathbf{e}$

Which uses so called J factors that lead to discount genetic information from genetic group covariate.

$$\mathbf{J} = \begin{bmatrix} -\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{1} \\ -\mathbf{1} \end{bmatrix} \begin{matrix} \text{ungenotyped} \\ \text{genotyped} \end{matrix}$$

Regular ssGBLUP model with genetic groups: which can give the same breeding values?


```
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

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

Requires computations:

$$\hat{\mathbf{a}}_d = \mathbf{Q}\hat{\mathbf{g}} + \hat{\mathbf{a}}$$

```
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_QP.dat
# iGFILE     LOWER data/iGL_w20_QP.dat
# iA22FILE   PEDIGREE

PEDFILE    data/sim_ped_mod.ped
PEDIGREE   ID am+p

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

PARFILE    data/AM.var

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

Gives directly $\hat{\mathbf{a}}_d$

```
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+p

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

PARFILE    data/AM.var

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

Gives $\hat{\mathbf{a}}_{\mathrm{QJ}}$

Hginv

Regular G inverse:

hginv_seq -lower -w 0.20 -Alower genot_mod.Lamat -a base_af_1000.dat -m PvR1 -c 2pq 9_Z0_id_16last.dat **iGL_w20.dat**

G inverse with QP part:

hginv_seq -lower -QP -P sim_ped_mod.ped -w 0.20 -Alower genot_mod.Lamat -a base_af_1000.dat -m PvR1 -c 2pq 9_Z0_id_16last.dat **iGL_w20_QP.dat**

So, what is going on?

We have 3 relationship parts influenced by genetic groups:

$$\begin{bmatrix} \mathbf{Q'A^{-1}Q} & \mathbf{-Q'A^{\cdot 1}} & \mathbf{-Q'A^{\cdot 2}} \\ \mathbf{-A^{1\cdot}Q} & \mathbf{A^{11}} & \mathbf{A^{12}} \\ \mathbf{-A^{2\cdot}Q} & \mathbf{A^{21}} & \mathbf{A^{22}} \end{bmatrix} + \begin{bmatrix} \mathbf{Q_2'G^{-1}Q_2} & \mathbf{0} & \mathbf{-Q_2'G^{-1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{-G^{-1}Q_2} & \mathbf{0} & \mathbf{G^{-1}} \end{bmatrix} - \begin{bmatrix} \mathbf{Q_2'A_{22}^{-1}Q_2} & \mathbf{0} & \mathbf{-Q_2'A_{22}^{-1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{-A_{22}^{-1}Q_2} & \mathbf{0} & \mathbf{A_{22}^{-1}} \end{bmatrix}$$

UPG can be implemented in alternative ways. Genetic groups for

1. $\mathbf{A}^{-1}$ only -- bad, should not be used
2. $\mathbf{A}^{-1}$ and $-\mathbf{A}_{22}^{-1}$ -- "altered QP H-inverse" or "partial QP" model by Q+J (regular G^{-1})
3. $\mathbf{A}^{-1}$ and $\mathbf{G}^{-1}$ and $-\mathbf{A}_{22}^{-1}$ -- full QP (option –QP in hginv)

When genetic groups are NOT included in the augmented genomic relationship matrix $\mathbf{G}_*^{-1}$
MiX99 works like case 2.

Different ssGBLUP models have it easy to use case 2 but difficult to make case 3.

There is some evidence that case 2 gives least bias.

Hginv has many options

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  1.0100
  PvR1a : PvR1 with diag(G) increased by 0.10000E-01
   PvR2 : P.VanRaden method 2 (-c m), singular if data allele frequencies
  PvR2m : PvR2 with diag(G) multiplied by  1.0100
   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
   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
     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.

-b mthd  : balancing G matrix with A22 matrix, mthd after option -b is
    ole : O.Christensen method of making G: G_n= b*G+a*J, where a&b by Ole
    own : Like "ole" but the coefficients are given, e.g., "-b own 0.99 0.01"
   PvR3 : P.VanRaden method 3: G_n= a*J + b*A22 where a&b by LS: G= a*J+b*A+E
    vit : Method by Vitezica et al.: G_a= G+a*J, a=1'(A22-G)1/(n^2)
   vit1 : G_1= G+J
-QP      : QP transformed unknown parent groups included in inv(G).
-P file  : input pedigree file for the option -QP or -JQ (input).
-UPG g   : maximum number of unknown parent groups is set to be g (default=500).
```

Lesser known is computation of SNP marker solutions:

```
Calculating marker solutions from DGV:
 -nt Ntrait : Number of traits. Default is 1.
 -u u_file  : Input u_file has DGVs of animals. Format: <id> <DGV>
 -ucol c    : DGV column(s). Default is 2 for trait 1.
 -iG iG_file: file iG_file has lower triangle of the inv(ZDZ'/k) matrix (input).
              File format is co-ordinate (default) or dense lower triangle (-lower).
 -k val : val was used in scaling the ZDZ' matrix when DGVs were calculated (see -u).
          When '-k data' is given, val is 2*sum(p*q) from the given genotype data
```

Be aware: scaling needs to be correct. This is error-prone.

ssGTABLUP

Assumes the **G** matrix has form: $\mathbf{G} = \mathbf{Z_c B Z_c'} + w\mathbf{A_{22}}$

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

ssGTeBLUP

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

where $\mathbf{Z}_c$ is centered marker matrix, ε is a small number like 0.01.

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

with the scaling constant $k = 2\sum_{i=1}^{m} p_i(1-p_i)$ and p_i are allele frequency for marker *i*.

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):

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

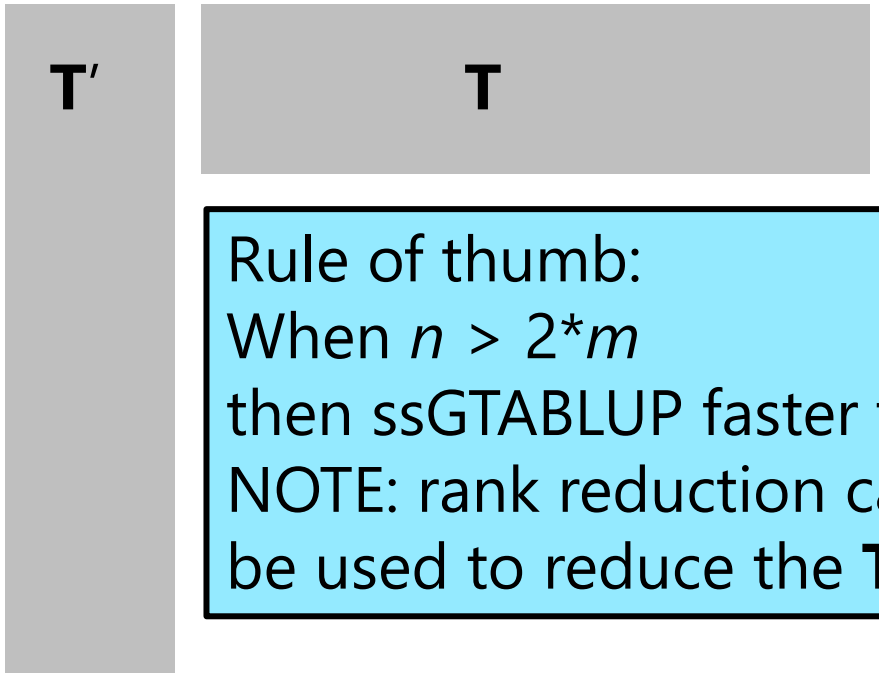
$$= \frac{1}{w}\mathbf{A}_{22}^{-1} - \mathbf{T}'\mathbf{T}$$

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

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

m = number of SNP markers

n = number of genotyped



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

There are two ways for ssGTABLUP in MiX99

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

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

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

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

- MiX99 has 2 component-wise approaches
  - Explicit: uses the centered marker matrix $\mathbf{Z}_c$ and $\mathbf{K}^{-1}$
  - Fully: uses $\mathbf{K}^{-1}$ and the original non-centered marker matrix,

    $\mathbf{Z}_c$ is built in-the-fly from marker data!
- Component-wise ssGTABLUP allows computing SNP marker solutions

T48eig_make and component-wise ssGTABLUP

`T48eig_make`		
`-nblk 1000`	:	1000 blocks in computations
`-nthr 10`	:	10 CPU threads
`-m dA`	:	scaling by tr(A22)/tr(G)
`-rpg 0.2`	:	residual polygenic proportion 20%
`-a base_af.dat`	:	allele frequencies for centering
`-FMT "(i10,26x,50240i1)"`	:	input format of genotypes
`-P mypedigree.ped`	:	pedigree file for A22 computations
`-F myinbreeding.inbr`	:	inbreeding coefficients (`-Fcol 3`)
`-ZC iC.bin`	:	`Output:` $\mathbf{K}^{-1}$ matrix for ICFILE
`mygenotypes.dat`	:	`Input:` genotype file
`Zc.bin`	:	`Output:` centered marker matrix **Z** for ZCFILE

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

Genetic groups to T matrix (not for component-wise ssGTABLUP):
-groups n : ssGTABLUP(old): include unknown parent groups (negative parent), n=maximum number of groups.

Luke

MiX99 example, ssGTABLUP

Traditional ssGTABLUP

```
...
TAFILE     TA.bin
iA22FILE  PEDIGREE

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

Explict 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
...
```

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 can be a minus sign.

RAM note: for data with almost 1M genotyped,
the explicit component-wise ssGTABLUP takes about 390GB RAM
but the fully component-wise takes about 76GB.

Improving the preconditioner: manual or automatic

Single-step uses $\mathbf{H}^{-1} = \mathbf{A}^{-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - \mathbf{A}_{22}^{-1} \end{bmatrix}$

Preconditioner of PCG often is based on diagonal elements of this matrix.

When the "iA22File PEDIGREE" command is given, the diagonal of $\mathbf{A}_{22}^{-1}$ needs to be computed explicitly or given.

If this information is not available → convergence suffers.

Two approaches:

1) automatic: let the preprocessor do the work (a Monte Carlo approach). This can take time for large data.

2) use a **calc_diag_iA22** program to calculate the diagonal:

```
calc_diag_iA22 -nthr 10 -MC 1000 -PAR -F my.inbr my.ped id_genotyped diA22.dat
```

The diA22.dat file has diagonal of $\mathbf{A}_{22}^{-1}$. However, a **precon.dat** file contains <ID code> <added value>

where <added value> is

- for ssGBLUP: minus diagonal of $\mathbf{A}_{22}^{-1}$
- for ssGTBLUP: needs (1/w-1) times the diagonal of $\mathbf{A}_{22}^{-1}$

In MiX99 CLIM file: `iHprecon precon.dat`

Further improving the preconditioner of ssGTABLUP

- Including only diagonal of inv(A22) may not be enough
- Including in T48eig_make option -dTT dTT_TA_w20.dat

allows computing diagonal of $\frac{1}{w}\mathbf{A}_{22}^{-1}\mathbf{Z}\left(\frac{1}{w}\mathbf{Z}'\mathbf{A}_{22}^{-1}\mathbf{Z}+\mathbf{B}^{-1}\right)^{-1}\mathbf{Z}'\frac{1}{w}\mathbf{A}_{22}^{-1}$

Then for 20% RPG:

paste **diA22.dat dTT_TA_w20.dat** | awk '{print $1, (1/.2-1)*$2-$4}' >diH_TAw20.dat

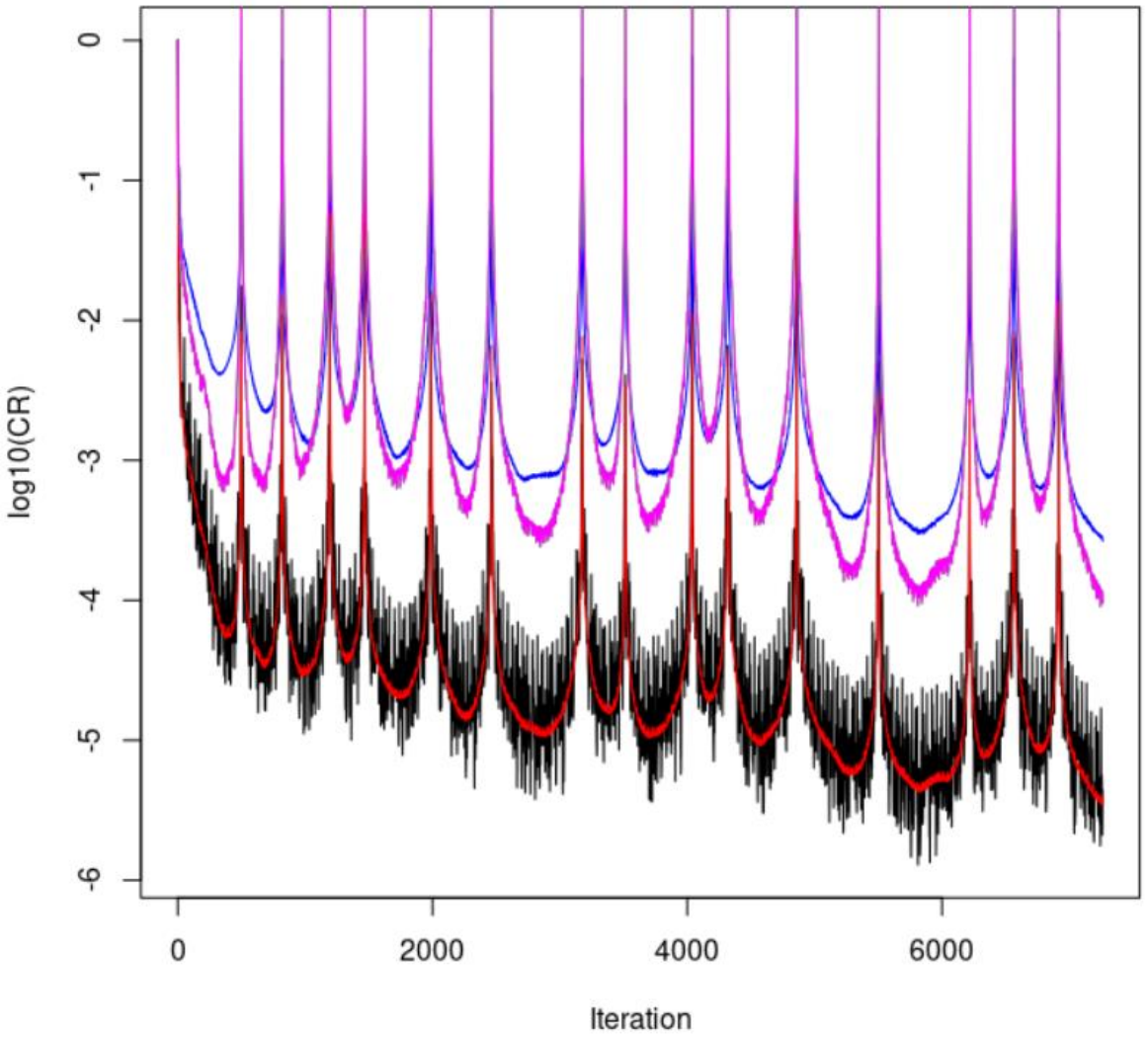
gives an efficient preconditioner (diagonal of $\mathbf{G}^{-1}$).

- However: the "-dTT" increases computations considerably.

There is seldom need for this complicated computations.

Sometimes convergence problems are a sign of a poor model and work should be done to make it better.

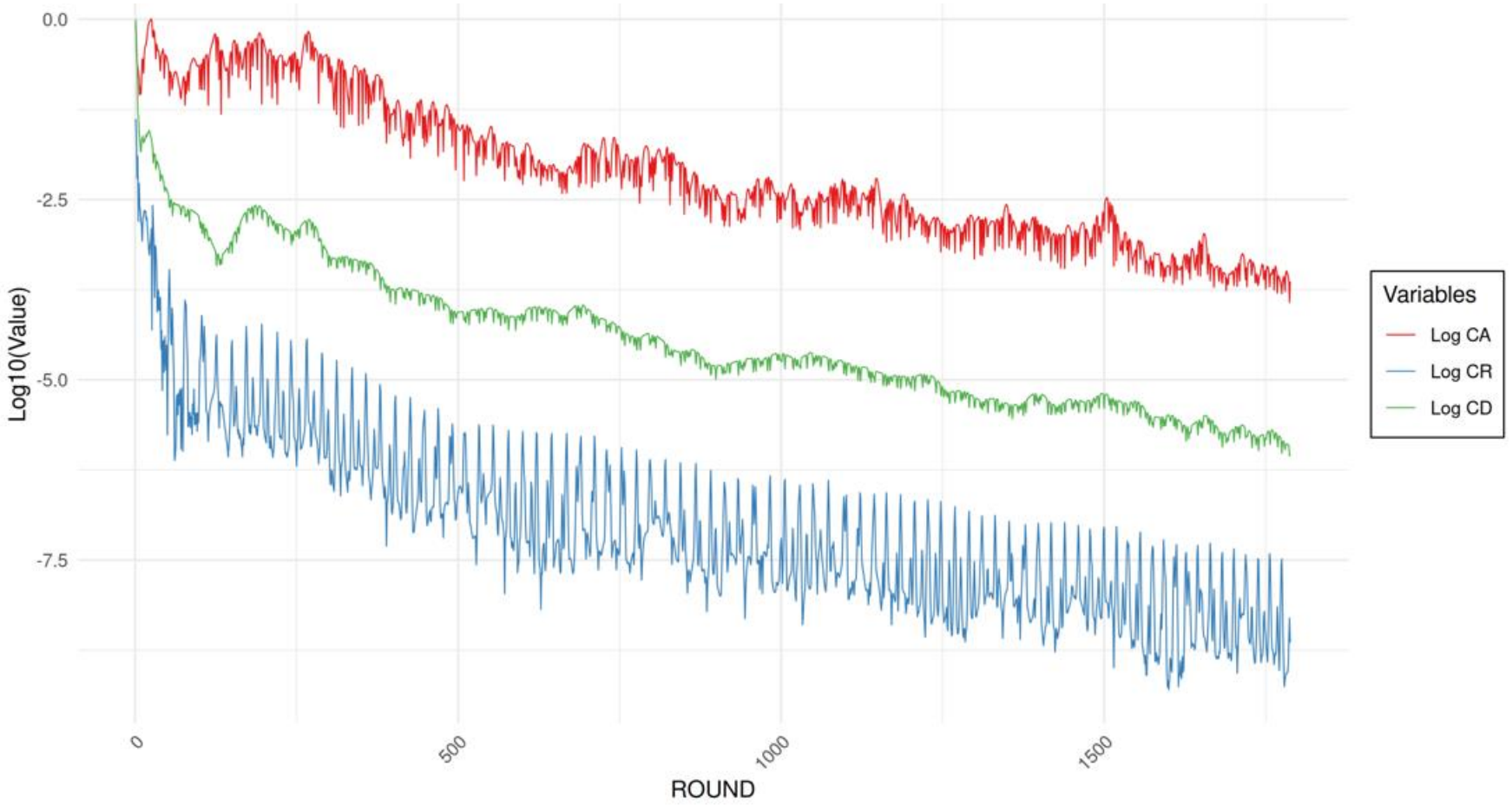
Poor convergence can be due to wrong model



Black = CR
Red = CM
Blue = CA
Magenta = CD

Reasons:
- Fixed unknown parent groups
- Genomic relationship matrix scale not good

After changes, the plots look nicer



ssSNPBLUP is an augmented version of ssGTABLUP

MME are

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{W} & \mathbf{0} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{W} + \mathbf{G}_0^{-1} \otimes \mathbf{H}_C^{-1} & -\mathbf{G}_0^{-1} \otimes \mathbf{K}_C \\ \mathbf{0} & -\mathbf{G}_0^{-1} \otimes \mathbf{K}_C' & \mathbf{G}_0^{-1} \otimes \mathbf{K} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{a}} \\ \hat{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{0} \end{bmatrix}$$

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

Note: 1) No **K** matrix is explicitly formed by MiX99

2) Genomic information is only in $\mathbf{K}_C$ and **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 vs ssSNPBLUP

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

Component-wise ssSNPBLUP with int-1 packed SNP-matrix:

```
SNPMATRIX  FIRST=2 LAST=50241 FORMAT='(i10,26x,50240i1)'  CENTER=p SCALE=p
SNPFILE    ../geno_data_nocand/ICBF_2018_10_genotypes_in_ped_ref.dat
CENTERFILE base_af_2col.dat
SSSNPBLUP  GTA 0.20
iA22File   PEDIGREE
```

No ZcFILE nor iCFILE

When number of markers is about 50K, the iCFILE means RAM increase of ~20 GB.

Large genotype information can take a lot of RAM

The size of the $\mathbf{Z}_c$ matrix file increases linearly with the number of genotyped.

In explicit component-wise ssGTABLUP, it becomes large as it has double precision numbers and is read to RAM.

The fully component-wise ssGTABLUP approach reads the original marker matrix and stores it to 1-byte integers

→ 1/8th of the RAM used by the $\mathbf{Z}_c$ matrix. The centering information is included iCFILE as before.

In **ssSNPBLUP**, no iCFILE but scaling information is needed.

→ uses even less RAM, but ssSNPBLUP has poorer convergence.

Packing 5 markers to a 1-byte integer is an option for huge data in fully component-wise ssGTBLUP and ssSNPBLUP.

MiX99 CLIM: ssGTABLUP and ssSNPBLUP

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 additional preprocessing

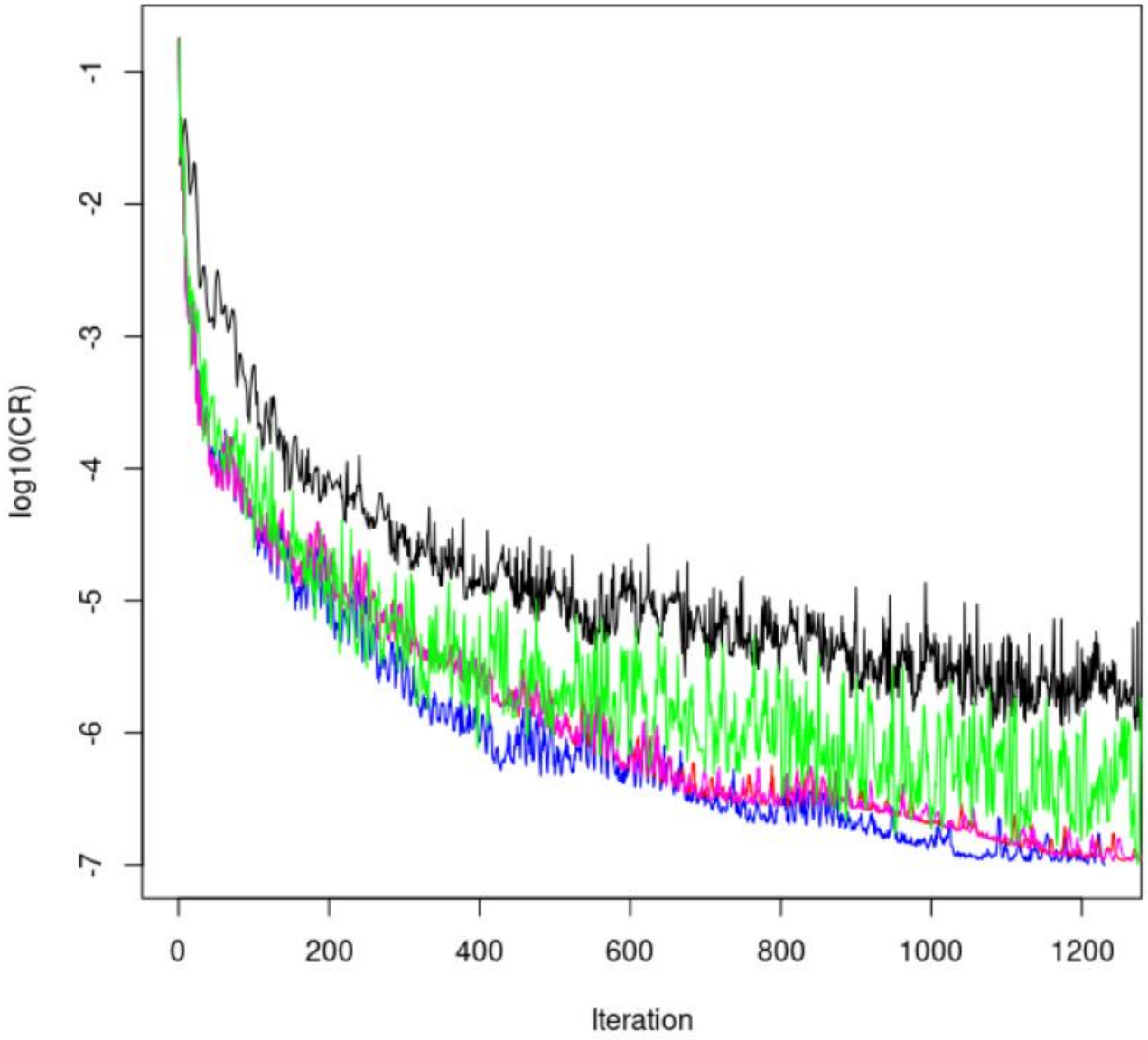
```
DATAFILE   ../data/9_SNP_WT_groups_2traits.dat
MISSING    -9
INTEGER    row ones ID
REAL       wt y y2 trueDGV wght g1 g2 g3 g4
DATASORT   PEDIGREECODE=ID

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 -
```

ssSNPBLUP convergence can be improved by second level preconditioner

Black = no sp
Green = sp 15
Blue = sp 45
Red = sp 60
Magenta = sp 100

ssGTABLUP uses about 650 iterations.



Single-step models

	Precomputed	Pre-program	RAM need (bytes)	SolSNP	Notes
ssGBLUP	$\mathbf{G}^{-1}$	hginv	$8n^2$	No	Small n
APY	$\mathbf{G}^{-1}$ of APY	hginv	$8n_cn$	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
Fully C-ssGTBLUP	$\mathbf{K}^{-1}$	T48eig_make	$8m^2+nm/5$	Yes	Medium to large n
ssSNPBLUP	-	-	$nm/5$	Yes	Very large n

C- = Componentwise

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

Convergence of ssGBLUP, APY, all ssGTBLUP are about the same.
Often ssSNPBLUP needs twice the number of iterations than those.

ssGTBLUP can use approximate approach by eigendecomposition of the T matrix. Small n large m cases.
Chromosomewise approximate approach by Odegaard et al. is also available but has been studied very little.

Luke

Metafounders

The use of metafounders is similar to any other single-step models except the need to estimate the gamma matrix and its use in the model.

Programs that can be used

Bpop

- Estimates base population allele frequencies
- Can be used to estimate the gamma matrix for the Metafounder model

RelaX2:

- Used to compute inbreeding coefficients
- Also when the gamma matrix is known for the Metafounder model

MiX99: hybrid parallel computing in mix99p

Mixed model equations:

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{W} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{W} + \lambda\mathbf{H}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix} \qquad \lambda = \frac{\sigma_e^2}{\sigma_a^2}$$

where the inverse of the relationship matrix is

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

Full or APY version of $\mathbf{G}^{-1}$
or **T** matrix
or componentwise **T**
or ssSNPBLUP.

Sparse parts: distributed memory approach using MPI

Option 'X' in mix99p: fastest but uses some more memory

Hybrid: MPI (distributed memory) and Cholmod/PARDISO library (shared memory parallel)

MPI is used when genomic matrix is not in memory,
ssGTBLUP with matrix in memory uses shared memory computing using MKL

Intel® Math Kernel Library

A word on marker weights

- Standard ssGBLUP assumes: marker variance equals the same genetic (co)variance matrix for all markers
- $\text{Var}(\mathbf{g}_i) = \mathbf{G}_0$, i = 1,…,m, where $\mathbf{G}_0$ is genetic (co)variance matrix.

- A general case assumes: $\text{Var}(\mathbf{g}_i) = \mathbf{G}_{0,i}$ is different covariance by marker.
- Estimating this matrix has many challenges.

A solution: estimate variances or weights for traits and assume their correlation is "one". We have a weighting matrix for each trait *i*: $\mathbf{D}_{ii}$

> Thus, $\mathbf{D}_{ij} = \left(\mathbf{D}_{ii}\mathbf{D}_{jj}\right)^{0.5}$. This will lead to a covariance structure that is equal to $\mathbf{V}_{g,k} = \mathbf{D}_{(k)}^{0.5}\mathbf{G}_0\mathbf{D}_{(k)}^{0.5}$, where the diagonal matrix $\mathbf{D}_{(k)}$ has the weights for all traits of marker *k*.

Full multi-trait case: ssSNPBLUP

MME for trait-specific marker weighted ssSNPBLUP:

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{W} & \mathbf{0} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{W} + \mathbf{G}_0^{-1} \otimes \mathbf{H}_C^{-1} & -\mathbf{G}_0^{-1} \otimes \mathbf{K}_C \\ \mathbf{0} & -\mathbf{G}_0^{-1} \otimes \mathbf{K}_C' & \mathbf{G}_0^{-1} \otimes \mathbf{Z}_c'\mathbf{C}^{-1}\mathbf{Z}_c + \mathbf{V}_g^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{a}} \\ \hat{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{0} \end{bmatrix}$$

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.

Weights are included in $\mathbf{V}_g$ which includes also the genetic covariance $\mathbf{G}_0$.

When no weights $\mathbf{V}_g^{-1} = \mathbf{G}_0^{-1} \otimes \mathbf{B}_w^{-1}$ where $\mathbf{B}_w = \mathbf{I}\frac{1-w}{s}$ ➔ standard ssSNPBLUP.

The $\mathbf{V}_g$ matrix is easy to invert because it is a block diagonal matrix (Liu et al. 2014) having blocks of size *T* for each marker *k*=1,…,*m*:

$$\mathbf{V}_{g,k} = \begin{bmatrix} \mathbf{g}_{0,11}\mathbf{B}_{11,k} & \mathbf{g}_{0,12}\mathbf{B}_{12,k} & \cdots & \mathbf{g}_{0,1T}\mathbf{B}_{1T,k} \\ \mathbf{g}_{0,21}\mathbf{B}_{21,k} & \mathbf{g}_{0,22}\mathbf{B}_{22,k} & \cdots & \mathbf{g}_{0,2T}\mathbf{B}_{1T,k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{g}_{0,T1}\mathbf{B}_{T1,k} & \mathbf{g}_{0,T2}\mathbf{B}_{T2,k} & \cdots & \mathbf{g}_{0,TT}\mathbf{B}_{TT,k} \end{bmatrix}$$

Heterogeneous variance or weights in ssSNPBLUP & MiX99


```
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
SNPParFile VC_het.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_2tr.var
TMPDIR     ./tmp

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

Heterogeneous variances for markers.

Trait specific marker weights with correlation of one for the covariance weights

```
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 DWEIGHT=T
SNPFILE    data/9_Z0_id_16last_nospace.dat
CENTERFILE data/base_af_1000.dat
SSSNPBLUP  GTA 0.20
WEIGHTFILE VR weights.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_2tr.var     # Variance component file
TMPDIR     ./tmp

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

MiX99 instruction code


```
DATAFILE        ../../data/data_mix99_calving.txt_good
INTEGER anim block sire dam damp byr sex hyDH hyDC hyBH hyBC hycalv dampar sirexh
REAL DH DC BH BC bsize cvig bwt mort1 mort2 gest agedam H0 FR JE M0 N0 RR ID DY1 DY2 AA AU BA BB CH HE LM PI PT SA SH SI IB damH0 damFR damJE damM0
damN0 damRR damID damDY1 damDY2 damAA damAU damBA damBB damCH damHE damLM damPI damPT damSA damSH damSI damIB FRxJEh FRxH0h H0xJEh DYxDYh TRxTRh C0
C0h 0Bx0Bh C0xDYh C0xTRh C0x0Bh DYxTRh DYx0Bh TRx0Bh

MISSING    -99
DATASORT   BLOCK=block PEDIGREECODE=anim

PEDFILE         ../../data/ICBF_6tr_pruned_mod.ped
PEDIGREE        G am
PARFILE         ../../data/covar_calv_6trt.txt

RANDOM          damp hyDH hyDC hyBH hyBC hycalv

SNPMATRIX  USE=PACK FIRST=2 LAST=50241 FORMAT='(i10,26x,50240i1)'  CENTER=p SCALE=p DWEIGHT=T
SNPFILE    ../../geno_data_nocand/ICBF_2018_10_genotypes_in_ped_ref.dat
CENTERFILE base_af_2col.dat
SSSNPBLUP  GTA 0.20
WEIGHTFILE SNPweights.dat
iA22File   PEDIGREE

INBRFILE   ../../data/ICBF_6tr_pruned.inbr
INBREEDING PEDIGREECODE=1 FINBR=3

TMPDIR          ./tmp

WITHINBLOCK G damp hycalv hyBC hyBH hyDC hyDH

PRECON b b b b b b b f

MODEL
DH    =  H0 FR JE M0 N0 RR ID   -   -  AA AU BA BB CH HE LM PI PT SA SH SI IB damH0 damFR damJE damM0 damN0 damRR damID   -      -     -      -
-     -     -     -     -     -     -     -     -     -     -   FRxJEh FRxH0h H0xJEh DYxDYh   -      -      -      -      -      -      -      -
  -     agedam            -        byr sex   -  hyDH  -     -     -       -   G(anim dam)
DC    =  H0 FR JE M0 N0 RR ID   -   -  AA AU BA BB CH HE LM PI PT SA SH SI IB damH0 damFR damJE damM0 damN0 damRR damID   -      -     -      -
-     -     -     -     -     -     -     -     -     -     -   FRxJEh FRxH0h H0xJEh DYxDYh   -      -      -      -      -      -      -      -
  -        -     gg(agedam|dampar) byr sex damp  -   hyDC   -     -       -   G(anim dam)
```

CLIM part of ssSNPBLUP with weights:


```
SNPMATRIX  USE=PACK FIRST=2 LAST=50241 FORMAT='(i10,26x,50240i1)'  CENTER=p SCALE=p DWEIGHT=T
SNPFILE    ../../geno_data_nocand/ICBF_2018_10_genotypes_in_ped_ref.dat
CENTERFILE base_af_2col.dat
SSSNPBLUP  GTA 0.20
WEIGHTFILE SNPweights.dat
iA22File   PEDIGREE
```

Genotypes in ICBF_2018_10_genotypes_in_ped_ref.dat (SNPFILE)

50240 marker columns (First=2, Last=50241),

Genotypes will be packed in RAM (USE=PACK),

Genotype marker centering by allele frequencies in base_af_2col.dat (CENTER=p, CENTERFILE),

Scaling by $k = 2\sum_{i=1}^{m} p_i(1 - p_i)$ (SCALE=p, CENTERFILE),

Single-step SNPBLUP with the residual polygenic proportion of 20% (SSSNPBLUP GTA 0.20),

Marker weights from the SNPweights.dat file (WEIGHTFILE SNPweights.dat) assumed to have a weight column for every trait and a row for every marker (DWEIGHT=T).

Estimating breeding value of newly genotyped candidate animals

predict_GEBV	
-a allele_frequencies_used.dat	Used for centering (same as in single-step done)
-P pedigree.ped	Pedigree from the single-step done
-F inbreeding.inbr -Fcol 3	Inbreeding coefficients from the single-step done
-DGV SolDGV_all	**Output: DGV for all individuals**
-RPGGEBV SolRPGEBV	Output: RPG part of the GEBV (often not interesting)
-GEBV SolGEBV_cand	**Output: GEBV of candidates**
-FMT "(i10,26x,50240i1)"	if format needed
genotypes_of_candidates.dat	Input: genotypes of the candidates
genotypes_in_single_step.dat	Input: from original single-step having all
Solani_of_single_step	
SolSNP_of_single_step	

Questions?