Unknown parent groups, convergence and indirect predictions in single-step

MiX99 course on genomic prediction

COURSE DAY 1, March 9th, 2026

Contents

- Genetic groups and single-step models
- Preconditioner & the second-level preconditioner
- Assessing convergence (by plots)
- predict_GEBV for candidate prediction
- A word on marker weights

Genetic groups and single-step models, or know what you are doing

Genetic groups in a model

Genetic groups can be included using

- Regression coefficients as unknown parent contributions or
- Using QP transformation as unknown parent groups (UPG)

Consider a single-trait ssGBLUP model with *genetic group regression coefficients*

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

where

$\mathbf{b}$ is a vector of fixed effects,

$\mathbf{g}$ is an r by 1 vector of random genetic group regression effects,

$\mathbf{Q}$ is a q by r matrix of known coefficients,

$\mathbf{u}$ is a q by 1 vector of random additive genetic effects,

and $\mathbf{e}$ is random residual vector.

Mixed model equations

MME for the ssGBLUP model are

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

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 $\hat{\mathbf{u}}_d = \mathbf{Q}\hat{\mathbf{g}} + \hat{\mathbf{u}}$.

Needs **Q** matrix in the model and in post-processing!
➔ QP transformation allows a simple set of MME with the need for post-processing.

In MiX99, the transformation is made by giving '+p' in the PEDIGREE command.
→ No need to give genetic groups by regression.

Luke

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

QP transformation ➔

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

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

The matrix parts containing genetic groups in the MME are

$$\begin{bmatrix} \mathbf{Q}'\mathbf{A}^{-1}\mathbf{Q} & -\mathbf{Q}'\mathbf{A}^{\cdot 1} & -\mathbf{Q}'\mathbf{A}^{\cdot 2} \\ -\mathbf{A}^{1\cdot}\mathbf{Q} & & \\ -\mathbf{A}^{2\cdot}\mathbf{Q} & & \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_2'\mathbf{G}^{-1}\mathbf{Q}_2 & \mathbf{0} & -\mathbf{Q}_2'\mathbf{G}^{-1} \\ \mathbf{0} & & \\ -\mathbf{G}^{-1}\mathbf{Q}_2 & & \end{bmatrix} - \begin{bmatrix} \mathbf{Q}_2'\mathbf{A}_{22}^{-1}\mathbf{Q}_2 & \mathbf{0} & -\mathbf{Q}_2'\mathbf{A}_{22}^{-1} \\ \mathbf{0} & & \\ -\mathbf{A}_{22}^{-1}\mathbf{Q}_2 & & \end{bmatrix}$$

Luke

Alternative Q matrix definitions lead to different MME after QP transformation

$$\begin{bmatrix} \mathbf{Q}'\mathbf{A}^{-1}\mathbf{Q} & -\mathbf{Q}'\mathbf{A}^{\cdot 1} & -\mathbf{Q}'\mathbf{A}^{\cdot 2} \\ -\mathbf{A}^{1\cdot}\mathbf{Q} & \mathbf{A}^{11} & \mathbf{A}^{12} \\ -\mathbf{A}^{2\cdot}\mathbf{Q} & \mathbf{A}^{21} & \mathbf{A}^{22} \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_2'\mathbf{G}^{-1}\mathbf{Q}_2 & \mathbf{0} & -\mathbf{Q}_2'\mathbf{G}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{G}^{-1}\mathbf{Q}_2 & \mathbf{0} & \mathbf{G}^{-1} \end{bmatrix} - \begin{bmatrix} \mathbf{Q}_2'\mathbf{A}_{22}^{-1}\mathbf{Q}_2 & \mathbf{0} & -\mathbf{Q}_2'\mathbf{A}_{22}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{A}_{22}^{-1}\mathbf{Q}_2 & \mathbf{0} & \mathbf{A}_{22}^{-1} \end{bmatrix}$$

- UPG can be implemented in alternative ways:

1.	$\mathbf{A}^{-1}$ only	-- very crude approximation, not recommended
2.	$\mathbf{A}^{-1}$ and $-\mathbf{A}_{22}^{-1}$	-- altered QP H-inverse
3.	$\mathbf{A}^{-1}$ and $\mathbf{G}^{-1}$ and $-\mathbf{A}_{22}^{-1}$	-- full QP

Beware of which kind of **Q** matrix or approach has been taken.

It is easy to include genetic groups as UPG in the pedigree and "augment" the $\mathbf{A}^{-1}$ matrix by genetic groups.

Including genetic group for the genomic part can involve a computational penalty.

Masuda Y, Tsuruta S, Bermann M, Bradford HL, Misztal I. Comparison of models for missing pedigree in single-step genomic prediction. J Anim Sci. 2021. https://doi.org/10.1093/jas/skab019.

Koivula, M., Strandén, I., Aamand, G. P., and Mäntysaari, E. A. (2022). Accounting for missing pedigree information with single-step random regression test-day models. *Agriculture* 12, 388. doi:10.3390/agriculture12030388

MiX99, QP and single-step models (CLIM: PEDIGREE … am+p)

	Altered QP H inverse	Full QP
ssGBLUP	Include no genetic groups in $\mathbf{G}^{-1}$	Include genetic groups in $\mathbf{G}^{-1}$
ssGTABLUP	Include no genetic groups in **T**	Include genetic groups in **T**
Component-wise ssGTBLUP	Default	Poorly tested option -fQP in mix99s
ssSNPBLUP	Default	Not available

hginv-options for $\mathbf{G}^{-1}$

```
-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).
```

T48eig_make-options for **T**

```
-groups n    : ssGTABLUP(T only): include unknown parent groups (negative parent), n=maximum number of groups.
-P ped_file : ssGTABLUP: pedigree file (input)
```

In both cases:
- In ssGBLUP/ssGTABLUP, unknown parent groups augment the matrix (either $\mathbf{G}^{-1}$ or **T**).
- Negative parent number in the pedigree file is an unknown parent group number.

Luke

Hginv

Regular $\mathbf{G}^{-1}$:

hginv_para -lower -w 0.20 -Alower genot.Lamat -a base_af.dat -m PvR1 -c 2pq my_geno.dat **iGL_w20.dat**

$\mathbf{G}^{-1}$ with genetic groups by QP:

hginv_para -lower **-QP -P my.ped** -w 0.20 -Alower genot.Lamat -a base_af.dat -m PvR1 -c 2pq my_geno.dat **iGL_w20_QP.dat**

T48eig_make

Regular **T**:

T48eig_make -rpg 0.2 -c 2pq -a base_af_1col.dat -P my.ped -F my.inbr -Fcol 3 my_geno.dat TA_w20.dat

T with genetic groups by QP:

T48eig_make -rpg 0.2 -c 2pq **-groups 50** -a base_af_1col.dat -P my.ped -F my.inbr -Fcol 3 my_geno.dat TA_w20_QP.dat

Regular ssGBLUP model with genetic groups

```
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
```

Which can give the same breeding values?

And which is correct?

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 full QP $\hat{\mathbf{a}}_d$

```
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 altered QP H inverse $\hat{\mathbf{a}}_d$

Preconditioner & the 2nd level preconditioner

Luke

Preconditioner

- Preconditioner for the iterative solver is given with the PRECON command
  - Example: `PRECON b d d b`
  - Each letter ‘b’ is a block diagonal preconditioner and ‘d’ is a diagonal preconditioner for a group of effects like across block fixed or random effects.

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

Preconditioner often uses 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.

- NOTE: “iA22FILE PEDIGREE” command will be used automatically when giving SSGBLUP & SSGTBLUP commands.

Single-step preconditioner: manual or automatic

Two approaches:

1) Automatic: let the preprocessor do the work (a Monte Carlo approach). **← RECOMMENDED: default**

2) Provide the update in a separate file calculated by for example **calc_diag_iA22** program:

```
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

- Sometimes including the diagonal of $\mathbf{A}_{22}^{-1}$ may not be enough

- For SSGTBLUP, T48eig_make has option -dTT my_TA_file.dat

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

Assuming ssGTABLUP with 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. THUS: NOT YOUR FIRST CHOICE!

There is seldom need for this complicated computations.

Convergence problems can be a sign of a poor model and work should be done to make the model better.

Second-level preconditioner

- Models having SNP marker data tend to have poor convergence
  - ssSNPBLUP model (by SNPMATRIX)
  - SNPBLUP model (by REGMATRIX)
- A second-level preconditioner often helps
  - Single-step with marker effects
  - A model with correlated REGMATRIX effects: SNPBLUP with a polygenic effect

- The second-level preconditioner is an option in the solver
  - mix99s … -sp 100 …

  Uses values 1/100 as the second level preconditioner.

  This is applied to all SNPMATRIX and REGMATRIX marker effects in the model.

- Typically, good values are from 20 to 200 depending on the data and variance components etc.
  - An example will be shown

Luke

Vandenplas J, Calus MPL, Eding H, Vuik C. A second-level diagonal preconditioner for single-step SNPBLUP. Genet Sel Evol. 2019;51:30.

Assessing convergence (by plots)

Convergence statistics

- Printed to screen and Conlog file by the solver

CA $$ca_{(k)} = \sqrt{\frac{\left(\boldsymbol{r} - \boldsymbol{C}\widehat{\boldsymbol{a}}^{(k)}\right)^T \left(\boldsymbol{r} - \boldsymbol{C}\widehat{\boldsymbol{a}}^{(k)}\right)}{\left(\boldsymbol{r}_a\right)^T \left(\boldsymbol{r}_a\right)}},$$

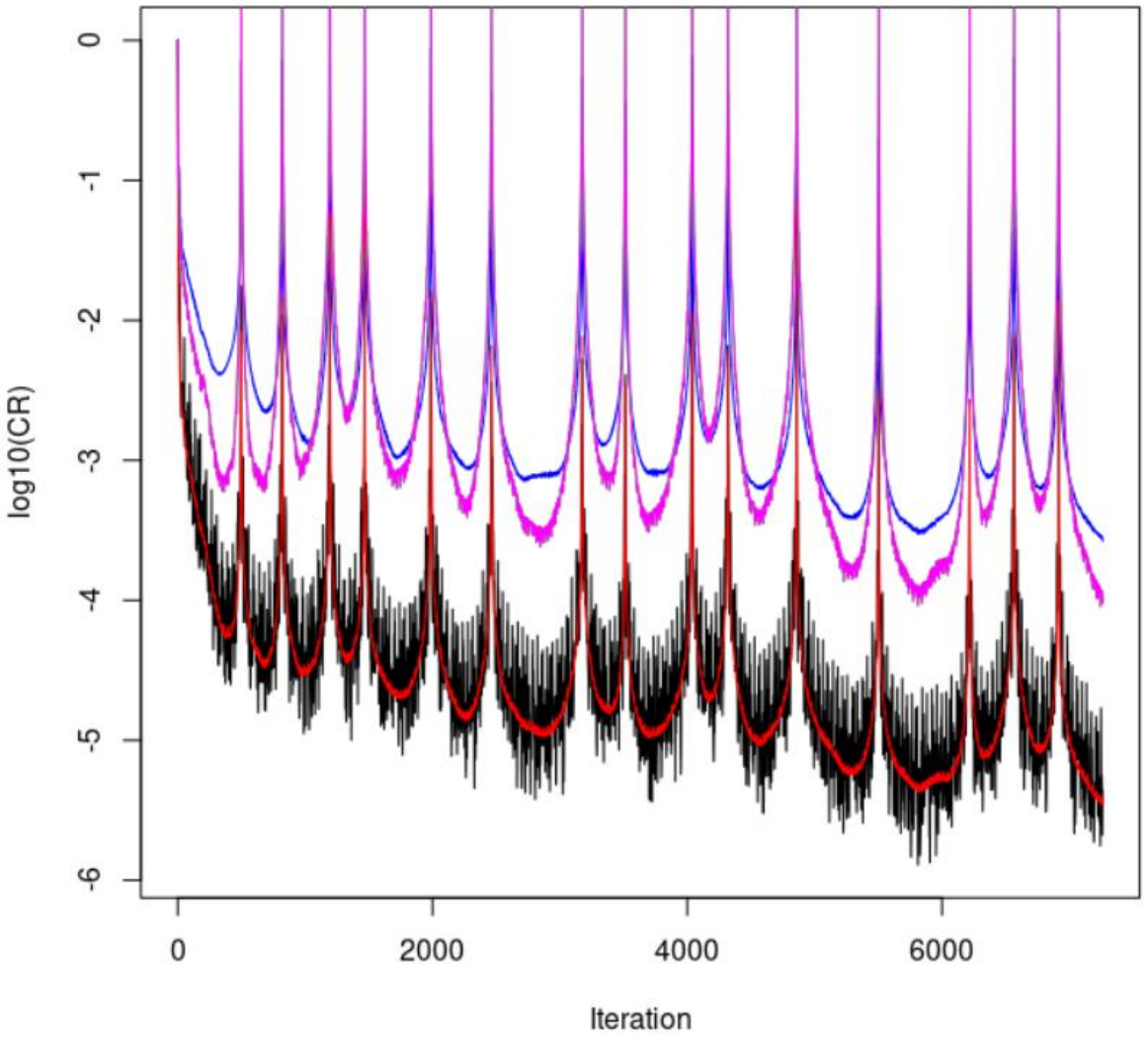
CR $$cr_{(k)} = \sqrt{\frac{\left(\boldsymbol{r} - \boldsymbol{C}\widehat{\boldsymbol{s}}^{(k)}\right)^T \left(\boldsymbol{r} - \boldsymbol{C}\widehat{\boldsymbol{s}}^{(k)}\right)}{\left(\boldsymbol{r}\right)^T \left(\boldsymbol{r}\right)}},$$

CM $$cm_{(k)} = \sqrt{\frac{\left(\boldsymbol{r} - \boldsymbol{C}\widehat{\boldsymbol{s}}^{(k)}\right)^T \boldsymbol{M}^{-1} \left(\boldsymbol{r} - \boldsymbol{C}\widehat{\boldsymbol{s}}^{(k)}\right)}{\left(\boldsymbol{r}\right)^T \boldsymbol{M}^{-1} \left(\boldsymbol{r}\right)}},$$

CD $$cd_{(k)} = \sqrt{\frac{\left(\widehat{\boldsymbol{s}}^{(k)} - \widehat{\boldsymbol{s}}^{(k-1)}\right)^T \left(\widehat{\boldsymbol{s}}^{(k)} - \widehat{\boldsymbol{s}}^{(k-1)}\right)}{\left(\widehat{\boldsymbol{s}}^{(k)}\right)^T \left(\widehat{\boldsymbol{s}}^{(k)}\right)}},$$

Monitoring convergence: logarithm of convergence statistic

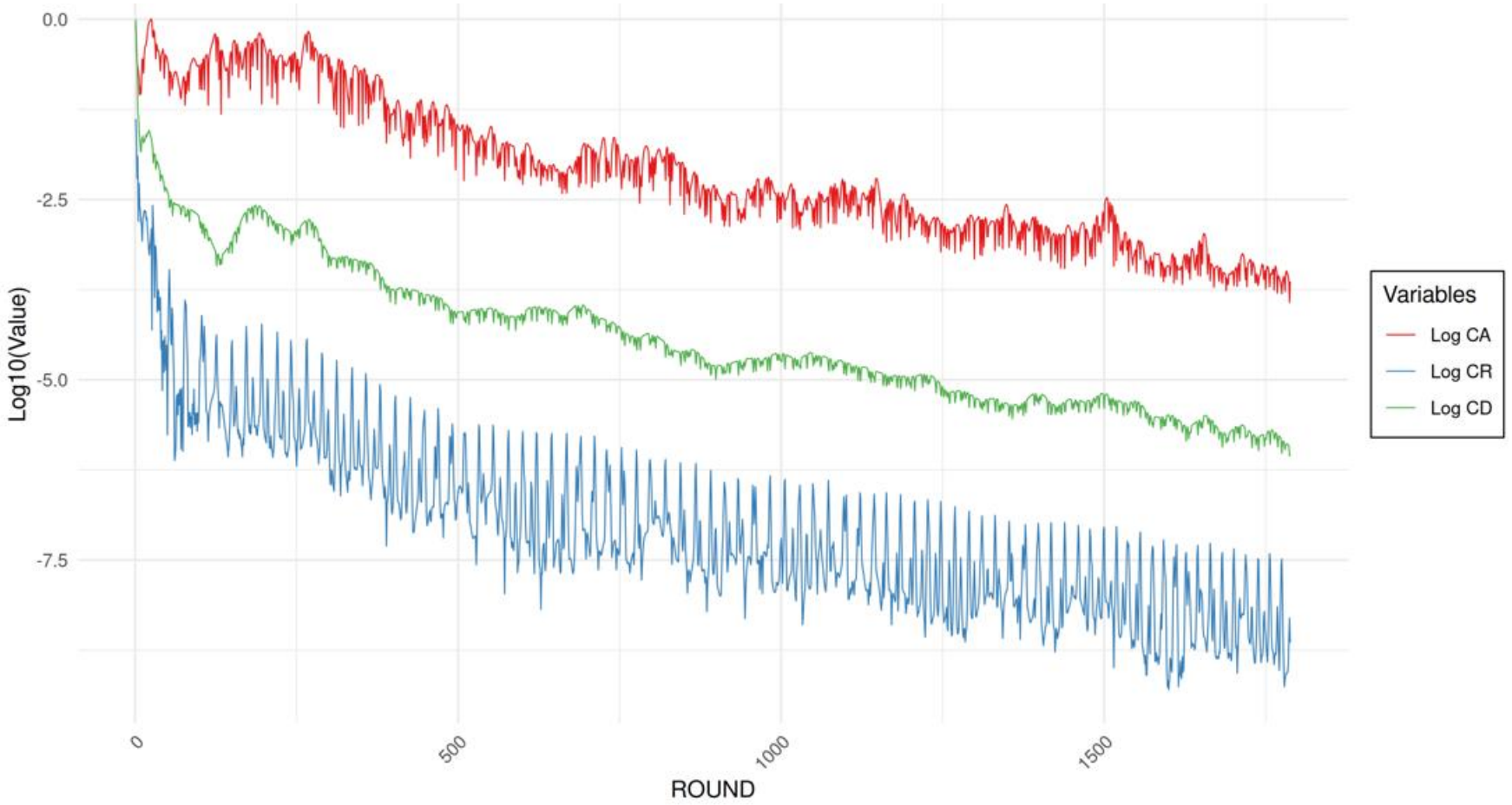
Poor convergence can be due to wrong model



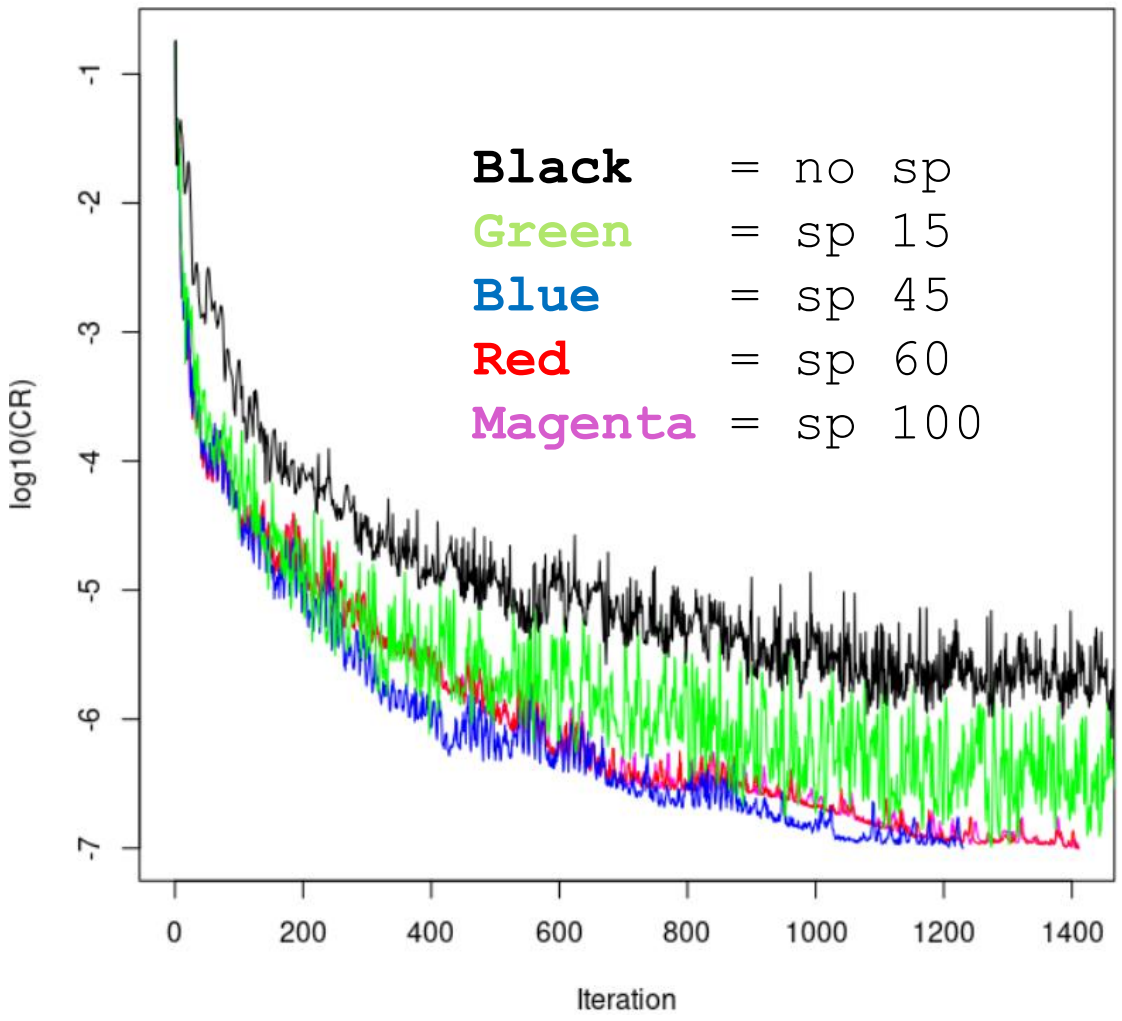
Black = CR
Red = CM
Blue = CA
Magenta = CD

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

After changes, the plots look nicer



ssSNPBLUP convergence improved by the second-level preconditioner



Second-level preconditioner (sp) values 60 and 100 are almost the same.

Value 45 looks the best for this data and model.

Memory options

Memory options in the solver

- Defaults usually work well
  - MEL for ssGBLUP, ssGTBLUP
  - MEA for the fully componentwise ssGTBLUP and ssSNPBLUP

- MEL: full matrix ($\mathbf{G}^{-1}$ or $\mathbf{T}$) read to memory
- MEA: packed (either 1 or 5 SNPs in byte) are unpacked to a block and used in the computations

- Change may be needed if memory use too high
  - MEB instead of MEA with a given block size

predict_GEBV for candidate prediction

Predict_GEBV work flow

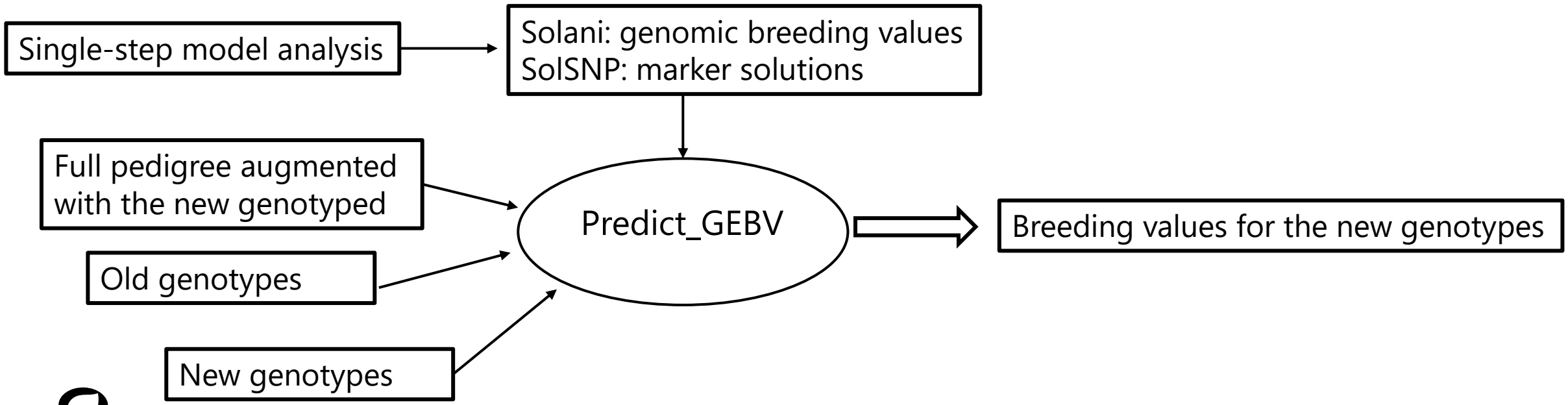
Backgroud:

Estimated breeding valules from a single-step models with a residual polygenic part have two parts:

Estimated genomic breeding value = Genomic breeding value + Residual polygenic breeding value

GEBV = DGV + RGEBV

➔ Knowing only the marker solutions is not enough when estimating breeding values for the newly genotyped individuals.
➔ Making the new analysis with some new genotypes can be too time consuming.
➔ Estimate the genomic breeding values of the newly genotyped using the latest evaluation data.



Luke

Vandenplas, J., J. Ten Napel, S. N. Darbaghshahi, R. Evans, M. P. Calus, R. Veerkamp, A. Cromie, E. A. Mäntysaari, and I. Strandén. 2023. Efficient large-scale single-step evaluations and indirect genomic prediction of genotyped selection candidates. Genet. Sel. Evol. 55:37. https://doi.org/10.1186/s12711-023-00808-z.

predict_GEBV options

```
USAGE:
prg_name [options] candGENOin refGENOin GEBVin SolSNPin

Options include

-info       : print these comments, current option values and stop.
-nthr n     : number of threads used in parallel computing is set to n.
-Zc         : assume refGENOin is in MiX99 ZcFile format (-FMT, -first, -m & -a apply candGENOin only).
-FMT FMT    : use format to read genotype files, eg. -FMT "(i9,1x,60000i1)".
-nospace    : no space between the marker genotypes but at least 1 space after the ID code.
-first c    : change default column number (2) of the first marker to be c.
-m method   : marker matrix centering method; affects used AF
    raw     : 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 centering markers,
              File has format: <allele frequency>
              First line is for the first marker, 2nd line for the 2nd marker ...
-mem low    : marker matrix not read to memory (default).
-mem high   : marker matrix read to memory.
-P ped_file : pedigree file.
-F F_file   : inbreeding coefficients file.
-Fcol c     : inbreeding coefficient column c.
-DGV DGVfile      : all genotyped individual DGV effects file (output).
-RPGGEBV RGEBVfile: reference individual RPG effects file (output).
-GEBV GEBVfile    : candidate GEBV file (output). Required!

Calculation methods (only one can be used):
-IOP        : use iteration on pedigree.
-IM         : use iteration in memory.
-CHM        : use Cholmod.
-PAR        : use MKL PARDISO (default).
```

Defaults are often good enough. Pedigree and output files need to be given through options!

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)"	when format needed
genotypes_of_candidates.dat	Input: genotypes of the candidates
genotypes_in_single_step.dat	Input: from the original single-step analysis
Solani_of_single_step	Input: from the original single-step analysis
SolSNP_of_single_step	Input: from the original single-step analysis

Some cautionary remarks

- The approach in predict_GEBV has been derived and designed to work for newly genotyped individuals
  - that have no progeny

- There are some manual steps like preparing the augmented pedigree and new genotyped file
  - Important to check that the results look reasonable
  - For example:
    - GEBV = mu + beta*DGV + e prediction for the reference and the candidate sets look similar
    - GEBV = mu + beta*DGV + e vs. GEBV = mu + beta*RPG + e predictions (linear regression estimates and correlations) look logical with respect to the residual polygenic proportion: Low proportion means high correlation of GEBV and DGV

predict_GEBV: GEBV and DGV statistics: reference

Linear regression: GEBV = mu + beta*DGV + e

Trait	Cor(DGV, GEBV)	hat(mu)	hat(beta)
1	0.980	-0.004	1.066
2	0.985	-0.002	1.074
3	0.964	-0.017	1.057
4	0.955	-0.017	1.040

predict_GEBV: GEBV and DGV statistics: candidates

Linear regression: GEBV = mu + beta*DGV + e

Trait	Cor(DGV, GEBV)	hat(mu)	hat(beta)
1	0.984	-0.002	1.044
2	0.987	-0.001	1.054
3	0.974	-0.012	1.041
4	0.970	-0.012	1.030

A word on marker weights

A word on marker weights

Strandén I, Jenko J (2024) A computationally feasible multi-trait single-step genomic prediction model with trait-specific marker weights. Genet Sel Evol 56(1):58. https://doi.org/10.1186/s12711-024-00926-2

- Standard ssGBLUP assumes: marker variance equals the same genetic (co)variance matrix for all markers

$$\text{Var}(\mathbf{g}_i) = \mathbf{G}_0,\ i = 1,\ldots,m,\ \text{where } \mathbf{G}_0 \text{ is genetic (co)variance matrix.}$$

- 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*.

MME for trait-specific marker weighted ssSNPBLUP:

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

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.

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 variances for markers.

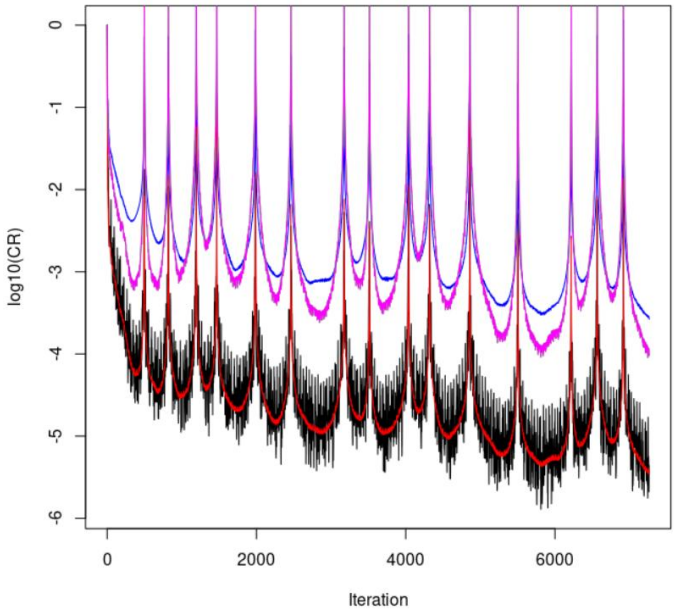
```
DATAFILE   ../data/9_SNP_WT_groups_2traits.dat
MISSING    -9
INTEGER    row ones ID
REAL       wt y y2 trueDGV wght g1 g2 g3 g4
DATASORT   PEDIGREECODE=ID

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
```

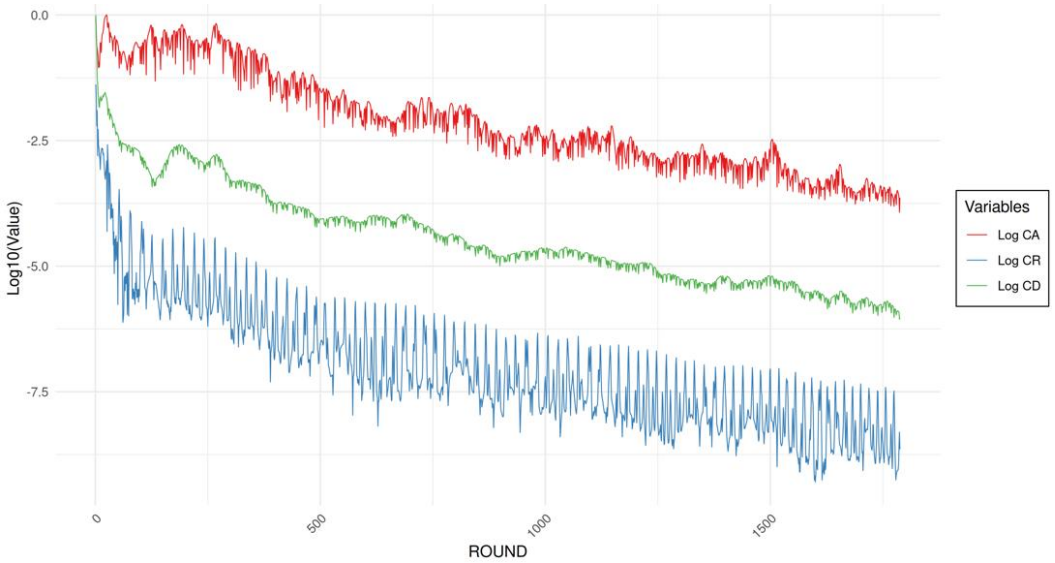
Trait specific marker weights with correlation of one for the covariance weights

```
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
```

Summary

Summary





Single-step models/data tend to require attention all the time

- small genomic data are most efficiently analyzed by different method than large data