Using snp_blup_rel to obtain genomic reliabilities

MiX99 course on genomic prediction

COURSE DAY 3, March 11th, 2026

Contents

- Motivation
- Theory
- Simple snp_blup_rel example
- Multiple single traits
- Reliabilities for the candidates
- Residual polygenic part?

Motivation

- Number of genotyped animals can be large
  - ➔ Making and inverting genomic relationship matrix becomes difficult
  - ➔ GBLUP becomes computationally too heavy
  - ➔ SNP-BLUP can be computationally less challenging

- Cost of computation of model reliabilities:
  - GBLUP: cubically by the number of genotyped individuals
  - SNP-BLUP: by the number of SNP markers, but requires many steps

➔**snp_blup_rel** for computing reliabilities

- computes prediction error variances (PEV) and corresponding reliabilities by SNP-BLUP
- computes elementary genomic breeding values by SNP-BLUP

Assume: genotypes in centered and scaled matrix $\mathbf{Z}$

- observations in vector $\mathbf{y}$
- weights (usually ERC/EDC) in the diagonal matrix $\mathbf{R}$

VanRaden I: $\mathbf{Z} = (\mathbf{M} - \mathbf{P})/\sqrt{k}$

SNPBLUP model: $\mathbf{y} = \mathbf{1}\mu + \mathbf{Z}\mathbf{g} + \mathbf{e}$

GBLUP model: $\mathbf{y} = \mathbf{1}\mu + \mathbf{u} + \mathbf{e}$

where it is assumed that $\mathrm{var}(\mathbf{g}) = \mathbf{I}\sigma_u^2$, $\mathrm{var}(\mathbf{u}) = \mathbf{Z}\mathbf{Z}'\sigma_u^2 = \mathbf{G}\sigma_u^2$, and $\mathrm{var}(\mathbf{e}) = \mathbf{R}\sigma_e^2$.

In SNPBLUP, the MME is $\lambda = \sigma_e^2 / \sigma_u^2$

$$\begin{bmatrix} \mathbf{1}'\mathbf{R}^{-1}\mathbf{1} & \mathbf{1}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{1} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \lambda\mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}} \\ \hat{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{1}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix}$$

The dimension of MME is 1+m, where m = the number of columns (markers) in the **Z** matrix.

In GBLUP, the MME is

$$\begin{bmatrix} \mathbf{1}'\mathbf{R}^{-1}\mathbf{1} & \mathbf{1}'\mathbf{R}^{-1} \\ \mathbf{R}^{-1}\mathbf{1} & \mathbf{R}^{-1} + \lambda\mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{1}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{R}^{-1}\mathbf{y} \end{bmatrix}$$

The dimension of MME is 1+n, where n = the number of rows (animals) in the **Z** matrix.

$\mathrm{PEV}(\hat{\mathbf{u}}) = \mathrm{var}(\hat{\mathbf{u}} - \mathbf{u}) = \mathbf{C}^{\mathrm{u,u}}\sigma_e^2$, where $\mathbf{C}^{\mathrm{u,u}}$ is sub-matrix of $\hat{\mathbf{u}}$ in the inverse of the MME matrix.

Also, $\mathrm{PEV}(\hat{\mathbf{u}}) = \mathrm{PEV}(\mathbf{Z}\hat{\mathbf{g}}) = \mathrm{var}\left(\mathbf{Z}(\hat{\mathbf{g}} - \mathbf{g})\right) = \mathbf{Z}\mathbf{C}^{\mathrm{g,g}}\mathbf{Z}'\sigma_e^2$

Model reliability

G = **ZZ**′ is genomic relationship matrix (VanRaden 1)

Z = (**M** – 2**P**)/*k*

M = marker matrix, **P**= allele frequency matrix

k = scaling factor, $\sqrt{(2 \sum p_j(1-p_j))}$, p_j= allele frequency of marker *j*

GBLUP

Model reliability for animal i:

$$1 - \{\mathbf{C}^{uu}\}_{ii} / v_g$$

$\mathbf{C}^{uu}$ = submatrix of MME inverse for breeding values

SNP-BLUP

Model reliability for animal i:

$$1 - \mathbf{Z}_i \mathbf{C}^{gg} \mathbf{Z}'_i / v_g$$

$\mathbf{C}^{gg}$ = submatrix of MME inverse for marker effects

$\mathbf{Z}_i$ = row in **Z** for animal *i*

SNP-BLUP requires additional computations after PEV has been computed

$v_g = \mathbf{G}_{ii} \sigma_u^2$

$\mathbf{G}_{ii}$ is diagonal element of animal *i* in **G** matrix, and

σ_u^2 is genetic variance.

Simple snp_blup_rel examples

Steps in snp_blup_rel

Read in observation weights	When no weights, this step is skipped.
Read in genomic data	Internal format by memory options (memlow, memmed, memhigh…)
Allele frequencies (file or data)	
Make SNPBLUP MME	**Can be time consuming**
Invert MME matrix	
Write MME inverse to disk	If not requested, this step is skipped.
Compute reliabilities	**Can be time consuming**

There are some variations in these steps depending on the memory mode.

Parallel computing by multi-threading is used in many of the steps.

Luke

Running snp_blup_rel_para

Genotype file: geno_spaced.dat

Output file: rel_result_1.dat

Z matrix: VanRaden Method 1

```
Memory and precision option
 -memlowAs   : Low memory, genotypes of at most 5000 animals in memory at a time (default).
 -memlowAp   : Like -memlowAs but uses -pcalc.
 -memlow  N  : Low memory, genotypes of at most N animals in memory at a time.
 -hmemlow N  : like -memlow but high precision computations.
 -mem1A      : Medium memory, genotypes in in RAM(int-1), computations in blocks of 10000 animals.
 -mem1  N    : Medium memory, genotypes in in RAM(int-1), computations in blocks of N animals.
 -memmed N   : like -hmemlow but SNP data in RAM(int-1).
 -memhigh    : High memory&precision, SNP data (filein) in memory with high memory use.
 -memlarge   : Large memory, high precision, SNP data (filein) in memory with extra high memory use.
```

10 threads

```
snp_blup_rel_para -nthr 10 -memlow 10000 \
                  -h2 0.36 -m PvR1 -c 2pq \
                  genotypes_spaced.dat rel_result_1.dat >snp_blup_rel_1.logi
```

Line continuation

Marker data file and output file

```
-m method : Genomic data coding method, the method after -m is
    raw   : use genotype data as such.
    101   : 101 coding (-1,0,1), assuming original is 012 coding.
 center   : center coding, i.e. PvR1 without scaling by 2*sum(p*q).
   PvR1   : P. VanRaden method 1, by default uses data allele frequencies.
   PvR2   : P. VanRaden method 2, by default uses data allele frequencies.

-c kval   : scaling in G =ZZ'/kval matrix where kval is
    2pq   : divide by 2*sum(p*q), default for PvR1
      m   : divide by the number of markers, default for PvR2
     m2   : divide by (number of markers)/2
     dA   : multiply by trace(A22)/trace(G) (need Ffile)
    one   : average diagonal of G will be one (Forni et. al. GSE 2011)
     no   : no scaling, default for 101 and raw

-h2    values : heritabilities (as by -straits). NOTE: lambda is calculated to be (1-h2)/h2.
```

Example data output

```
Input genotype file   : genotypes_spaced.dat
Genotype values       : real numbers
Check 012 coding      : Yes (takes time!)
Genotypes in memory   : No, memlow, block size=       10000
High precision all    : No
Genotype reading      : parallel
Reliability           : partially parallel

Input data file       : NONE GIVEN (all assumed to have an observation).
Number of traits      :            1
Column of weight      : no weights
Mean in the model     : Yes
Variance ratio lambda:    1.77777777777778
Heritability       h2:  0.360

Output reliabilities : rel_result_1.dat

Method for   Z matrix: PvR1     method code=          7
             Z matrix: P.VanRaden 1
Scaling of Z         : divide by sqrt(2*sum(pq))
Allele frequencies   : estimated from the genotype data
Number of MKL threads:           10
Number of OMP threads:           10
```

It is good to check this summary table.

Good practice: check the table using the –info option:

`snp_blup_rel_para -info -nthr 10 …`

```
First 10 animals:
        id       obs.wt     PEV/var(e)  diag(G)      r2
 857337311         1.00     0.13521       0.9138  0.7370
 866880391         1.00     0.16333       0.9814  0.7041
 872247712         1.00     0.12036       0.9424  0.7730
 860903414         1.00     0.21833       1.0238  0.6209
 822316164         1.00     0.15106       0.9786  0.7256
 858167932         1.00     0.16368       1.2761  0.7720
 864327427         1.00     0.13818       0.9463  0.7404
 868449788         1.00     0.24127       0.9705  0.5580
 862752339         1.00     0.15402       1.0140  0.7300
 744293496         1.00     0.18704       0.9796  0.6606
```

Multiple single traits

Several single traits

- Many single traits can be done at the same time.
- These can have different weights and heritabilities

6 heritabilities

Data file

```
snp_blup_rel_para -nthr 10 -ignore -memlow 10000 -a base_af_2col.dat \
        -straits 6 -h2 0.3590 0.4554 0.4067 0.4524 0.5890 0.8271E-01 \
        -m PvR1 -wt 2 3 4 5 6 7 -o id_data_Solani_genotyped \
        -FMT '(i10,26x,50240i1)' genotypes_in_mydata_ref.dat rel_result_memlow.dat
```

6 single trait reliability computations

6 weight columns in the data file

```
Number of observations for genotyped indiv  :      435834
Number of records with non genotyped id     :           0
Number of records with non genotyped id(ped):           0

Number of records with weight > 0           :       26324  85373  42104 174760 204701  25215
Number of records with weight <= 0          :      409510 350461 393730 261074 231133 410619
Number of records with all trait weights 0  :           0
************************************************************
Note: Genotypes of records with weight <= 0 are
      NOT included in the marker effect PEV computations.
************************************************************

Minimum weight (weight>0)                   : 1.00000 1.00000 1.00000 1.00000 1.00000  20.0000
Maximum weight                              : 4.00000 4.00000 4.00000 4.00000 5.00000 100.0000
Average weight                              : 1.23678 1.16226 1.41801 1.24364 3.08366  41.7386
Number of records with negative weight      :  409510  350461  393730  261074  231133   410619
These weights were set to zero.
```

Continued…


```
Number of observations with weight >0 :       26324       85373       42104
     174760      204701       25215
Observation count matrix:
    1 :    26324
    2 :        0   85373
    3 :        0       0   42104
    4 :        0       0       0  174760
    5 :      440    1551   17726   83524  204701
    6 :     1540    3878    1784    6613   10941   25215
```

After making and inverting the MME matrix, the reliabilities are written to file

```
First 10 animals:
       id obs.wt obs.wt obs.wt obs.wt obs.wt obs.wt PEV/var(e) PEV/var(e)   PEV/var(e) PEV/var(e) PEV/var(e)   PEV/var(e)    diag(G)      r2      r2      r2      r2      r2      r2
857337311 1.00   0.00   0.00   0.00   0.00   0.00   0.12779      0.89522E-01 0.32966    0.19020    0.84418E-01 0.19412E-01   1.1064  0.7938  0.9032  0.5653  0.7919  0.9468  0.8054
864315414 0.00   1.00   0.00   0.00   0.00   0.00   0.14889      0.96671E-01 0.33252    0.18947    0.86114E-01 0.19582E-01   1.0953  0.7573  0.8945  0.5571  0.7906  0.9451  0.8017
866880391 2.00   0.00   0.00   0.00   0.00   0.00   0.13424      0.13511     0.36324    0.20009    0.10155     0.24225E-01   1.1198  0.7860  0.8557  0.5268  0.7837  0.9367  0.7601
872247712 1.00   0.00   0.00   0.00   0.00   0.00   0.11316      0.84516E-01 0.32903    0.18034    0.84620E-01 0.20280E-01   1.1364  0.8222  0.9111  0.5776  0.8079  0.9480  0.8021
867786501 0.00   1.00   0.00   0.00   0.00   0.00   0.18888      0.11030     0.35700    0.19461    0.99581E-01 0.21647E-01   1.0702  0.6849  0.8767  0.5133  0.7799  0.9351  0.7757
854391025 0.00   1.00   0.00   0.00   0.00   0.00   0.17781      0.10289     0.34844    0.20087    0.97484E-01 0.23026E-01   1.1102  0.7140  0.8892  0.5421  0.7810  0.9387  0.7700
753271022 0.00   2.00   0.00   0.00   0.00   0.00   0.21679      0.11074     0.38795    0.22336    0.11910     0.28709E-01   1.0968  0.6471  0.8793  0.4840  0.7535  0.9242  0.7097
865537117 0.00   1.00   0.00   0.00   0.00   0.00   0.21755      0.12198     0.38339    0.21997    0.11601     0.28080E-01   1.0644  0.6351  0.8630  0.4746  0.7499  0.9240  0.7074
857470824 0.00   1.00   0.00   0.00   0.00   0.00   0.19618      0.11328     0.36203    0.20090    0.10335     0.25390E-01   1.1000  0.6816  0.8768  0.5199  0.7789  0.9344  0.7440
866883298 0.00   1.00   0.00   0.00   0.00  50.00   0.18313      0.11229     0.32408    0.17755    0.89647E-01 0.10931E-01   1.0325  0.6833  0.8699  0.5421  0.7919  0.9394  0.8826
```

This data had 435834 genotyped individuals, each having 50240 markers. Peak RAM was ~68GB.

The single trait analysis used ~16GB → each new trait takes about ~10GB more memory (MME coefficient matrix+ some other).

Some time and peak RAM results by memory option and 3 cases: -FMT option

snp_blup_rel v. 1.07	A – time (min)	A- RAM (GB)	B- time (min)	B- RAM (GB)	C –time (min)	C- RAM (GB)
memlow (10K)	8.6	15.3	75	15.3	230	67.7
memlowAs (5K)	8.4	13.3	88	13.4	225	66.2
memlowAp (5K)	7.0	31.1	65	31.1	189	83.9
hmemlow (10K)	12.2	32.2	141	32.0	359	142.6
mem1 (10K)	6.9	16.6	65	38.4	210	90.8
mem1A (10K)	5.3	16.6	67	38.4	170	90.8
memmed	11.6	33.3	136	55.0	385	167.8
memlarge	14.2	40.0	118	298.5	279	1293.6
memhigh	10.9	42.7	82	216.8	226	766.2

A: 1 trait, 26,324 genotyped B: 1 trait, 435,834 genotyped

C: 6 traits, 435,834 genotyped

All : 50250 markers, weights

Some time and peak RAM results by memory option and 3 cases: -nospace option

snp_blup_rel v. 1.07	A – time (min)	A- RAM (GB)	B- time (min)	B- RAM (GB)	C –time (min)	C- RAM (GB)
memlow (10K)	5.5	15.3	54	15.3	191	67.7
memlowAs (5K)	5.6	13.3	53	13.4	200	66.1
memlowAp (5K)	5.5	31.1	48	31.2	218	83.9
hmemlow (10K)	9.4	32.2	106	32.3	316	138.1
mem1 (10K)	5.5	16.6	64	38.4	201	90.8
mem1A (10K)	5.5	16.6	54	38.4	200	90.8
memmed	9.3	33.3	95	55.2	328	163.5
memlarge	12.7	40.0	78	298.6	217	1292.6
memhigh	10.4	42.7	57	216.8	184	766.2

A: 1 trait, 26,324 genotyped B: 1 trait, 435,834 genotyped

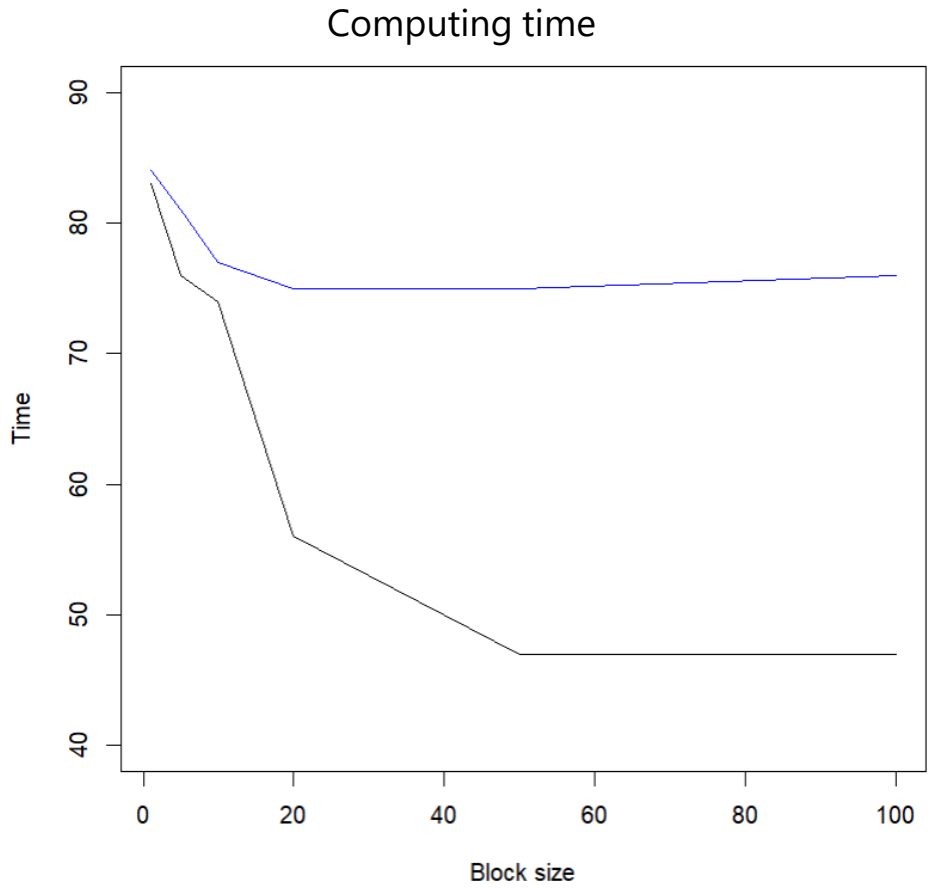
C: 6 traits, 435,834 genotyped

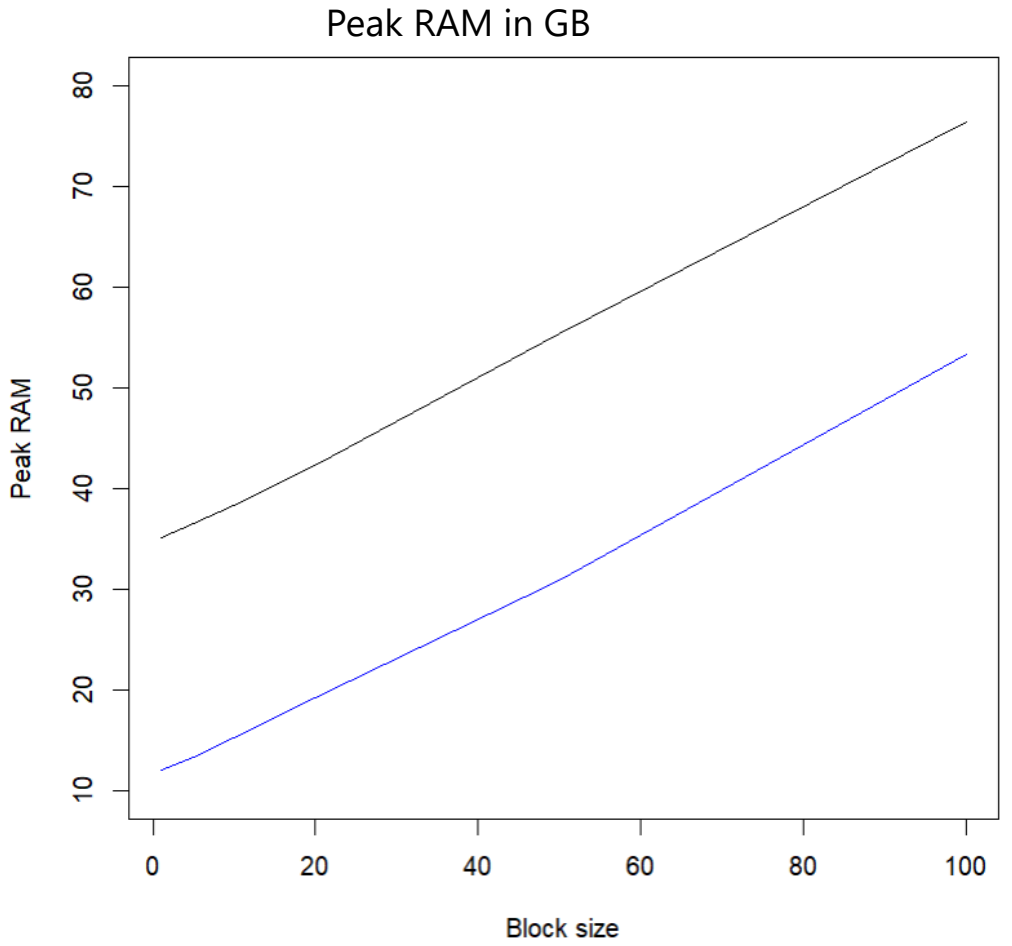
All : 50250 markers, weights

Luke

Options –memlow vs. –mem1 under different block sizes

Blue = Option –memlow, Black= Option –mem1

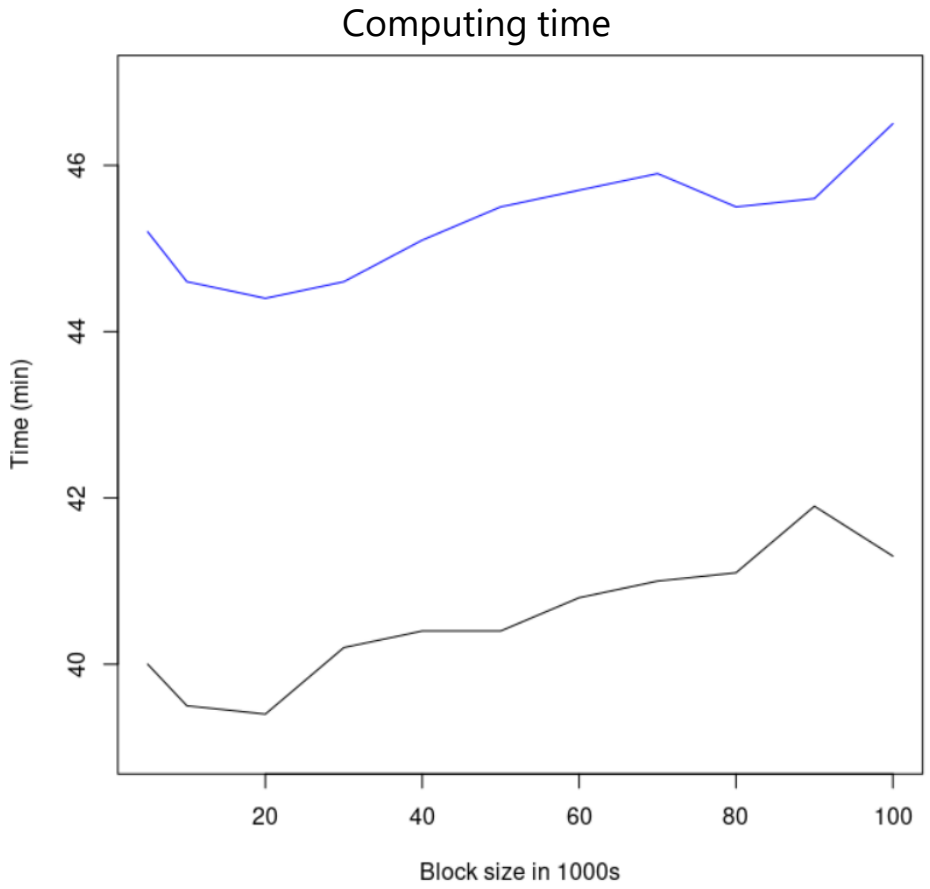


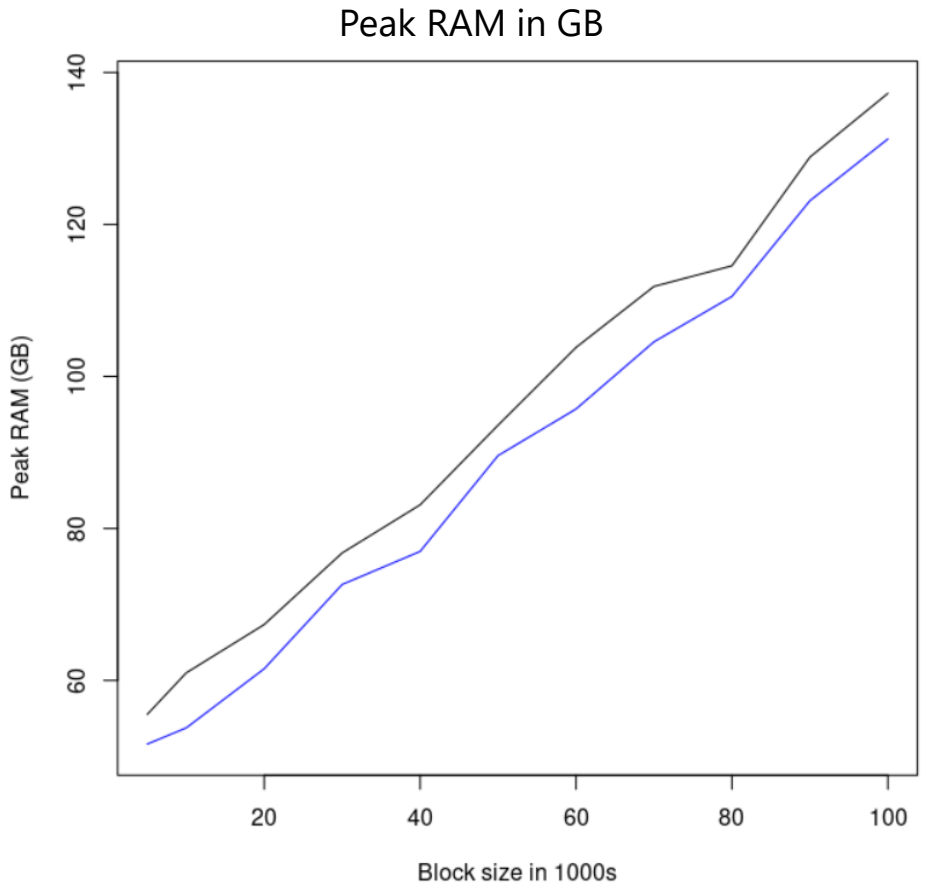


A quite low block size is enough for memlow, but mem1 may be faster with a larger block size.

Options –hmemlow vs. –mem1s under different block sizes

Blue = Option –hmemlow, Black= Option –mem1s in version 0.99





A quite low block size seems to be enough
Increased precision has a time penalty.
However, when heritability over 90% there can be a difference in results

Reliabilities for the candidates

Some options

- Entering options: command line (as shown) or in a file specified by the –f option.

  Or both: After –f option, all commands are read from the file

New genotypes arrive but give no new information to reliability computation

- No need to rerun snp_blup_rel using all data
- Use previous run inverted MME matrix to compute reliabilities of the candidates

Candidate reliabilities by using PEV from an earlier run using options:

```
-iCin  inv_MME_file : inverse of MME for SNP-BLUP read from file, txt/2R for text (input)
-iCout inv_MME_file : inverse of MME for SNP-BLUP written to file, txt/2R for text (output)
-Cout MME file      : MME for SNP-BLUP written to file in plain or 2R text format (output)
```

An example: reference and candidates

Step 1: compute reliabilities for the reference individuals and write the MME coefficient matrices to disk

```
snp_blup_rel_para -nthr 10 -ignore -memlow 10000 -a base_af_2col.dat -iCout iMME_memlow.bin \
                  -straits 6 -h2 0.3590 0.4554 0.4067 0.4524 0.5890 0.8271E-01 \
                  -m PvR1 -wt 2 3 4 5 6 7 -o id_data_Solani_genotyped \
                  -FMT '(i10,26x,50240i1)' genotypes_in_mydata_ref.dat rel_result_memlow.dat
```

Step 2: some time later candidate genotypes become available

```
snp_blup_rel_para -nthr 10 -ignore -memlow 10000 -a base_af_2col.dat -iCin iMME_memlow.bin \
                  -straits 6 \
                  -m PvR1    \
                  -FMT '(i10,26x,50240i1)' genotypes_in_mydata_cand.dat rel_result_memlow_cand.dat
```

Notes:

1) allele frequencies (-a) and method (-m) have to be the same for this to work
   - recall that the centered marker matrix has to be used in order to compute reliabilities
2) remember to use a different output file!
3) because heritabilities were already provided in Step1, they cannot be changed nor given again
4) candidates cannot have data to update the MME

Residual polygenic part?

Some other options

A so-called residual polygenic part can be included in the genomic matrix (= regularization $\mathbf{C}=w\mathbf{A}_{22}$).
The proportion is given by '-w'.
Exact computations use a pre-computed **A** matrix of the genotyped (-A or –Alower).
Approximate computations use Monte Carlo sampling: -MC or –fullMC.

```
-w   w   : relative proportion of RPG effect (A22 matrix) from total.
RPG effect by Monte Carlo sampling of the A22 matrix:
-ped file: pedigree file for residual polygenic effect (RPG).
-F Ffile : inbreeding coefficients file (input). Format: <id> <number> <F coefficient>
-Fcol fc : column (default 3) for the inbreeding coefficients in the -F file
-MC  n   : RPG uses Monte Carlo with n samples.
-RNG t   : Random number generator seed: t is D=Fortran default, G=given, R=clock.
-fullMC M: use Monte Carlo for marker effects as well.
           M is memory usage: low, medium or high.
-exact_dG: use exact diagonal of G/A22 in r2 formula. Default: MC approximation.
-exact_MC: lower level MC approximation of PEV and -exact_dG (experimental).
-AMCout f: output Monte Carlo generated estimate to the A22 matrix.

Alternative to RPG by Monte Carlo sampling of the A22 matrix:
-A Amat  : Amat file has pedigree based relationship matrix A22 (input)
           Each row has format: <id_1> <id_2> <relationship value>
-Alower Amat : Amat file has pedigree based relationship matrix A22 (input)
           Matrix is in lower triangle dense format.
```

This is often computationally too heavy.

Better use:

The RPG effects can be accounted for in the final reliability of GEBV by blending the Model (2) reliabilities with the reliabilities of the traditional EBV from PBLUP in Step 1 using the following equation:

$$r^2_{g,g,i} = \frac{(1-\omega)\mathbf{G}_{ii}r^{2*}_{g,g,i} + \omega\mathbf{A}_{22ii}r^2_{p,g,i}}{(1-\omega)\mathbf{G}_{ii} + \omega\mathbf{A}_{22ii}},$$

where $\mathbf{A}_{22ii}$ is the diagonal element i of the $\mathbf{A}_{22}$ matrix which is equal to $1 + \mathrm{F}_i$ with F_i equal to the pedigree-based inbreeding coefficient of animal i.

Summary

Summary

- Options for the "**G**" matrix: scaling and centering (-m & -c)
  - related option: -a for allele frequencies
- Marker matrix formats: default, -FMT, -nospace, -int, -real
  - Can select only some markers by the –s option (may inflate r^2)
- Different memory models: from memlow to memhigh
  - memlow often best
- Monte Carlo for the residual polygenic part may not be a good idea
- Multi-threading is used in many steps
  - Choose the number of threads well
- Several single traits in the same go: -straits with -h2 & other options
  - Be aware that every trait increases memory need

Luke


Thank you!
© Luke