MTG (Multi-trait GREML and GBLUP program)

Version 1.3
SH Lee (Jul/14)

MTG is the computer program implementing a multivariate linear mixed model to fit complex covariance structures that can be constructed based on genomic information, i.e. multivariate version of GCTA REML[1]. It gives residual maximum likelihood (REML) estimates for genetic and environmental variance and covariance across multiple traits. It estimates best liner unbiased prediction (BLUP) for quantifying genetic merits or genetic risk. MTG uses the direct average information algorithm[2] .

Citation:
If you use MTG software, please cite the following paper.

Maier, R., et al. Joint analysis of psychiatric disorders increases accuracy of risk prediction for schizophrenia, bipolar disorder and major depression disorder. *The American Journal of Human Genetics* **96**, 283-294 (2015)

1. MTG Command

The command for MTG is easy and a simple modification of the PLINK[3] and GCTA[1] syntax.

```
./mtg1.3 -p {plink fam file name} -d {phenotype file name} –g {grm file name} -cc
{class covariate file name} –qc {continuous covariate file name} -out {output file
name} -sv {starting value file name} –mod {number of traits}
```

E.g.,
```
./mtg1.3 -p test.fam -d test.dat -g test.grm -cc test.cov -qc test.pc -out test.out -
mod 5
```

<fam file for -p>
The PLINK fam file is your *.fam file that used in estimating the grm.

<grm file for -g>
For the grm file, you should unzip the .gz file from GCTA, delete the third column. Then, it looks like

```
1 1 0.999
2 1 0.011
2 2 1.031
3 1 0.02
…..
```

For the GCTA *.gz file for grm, the following command will do it easily,

```
zcat test.grm.gz | awk {print $1,$2,$4}' > test.grm
```

<binary grm file for -bg>
MTG can read binary GRM files generated by GCTA.
E.g., ./mtg1.3 -p test.fam -d test.dat -bg test.grm.bin -cc test.cov -qc test.pc -out test.out -mod 5

<phenotype file for -d>
With 5 traits model, the columns have FID, IID, t1, t2, t3, t4 and t5 (phenotypes for trait 1 ~ 5). It looks like

```
1 1 0.02 0.71 -0.02 0.04 -0.62
1 2 0.12 0.31 -0.27 NA -0.35
2 1 0.22 0.25 -0.28 0.63 -0.15
……
```

Missing values should be coded as NA.

<files for –cc (class covariate) and –qc (continuous covariate)>
The FID and IID order for phenotype file, covariate files (cc, qc) should be the same.
Missing values should be coded as NA.

2. MTG Extra options

-cove 1: parameterising residual covariance
E.g. ./mtg1.3 -p test.fam -d test.dat -g test.grm -cc test.cov -qc test.pc -out test.out -mod 5 -cove 1

-thread k: k paralleled computation
E.g. ./mtg1.3 -p test.fam -d test.dat -g test.grm -cc test.cov -qc test.pc -out test.out -mod 5 -thread 10

-sv {file name}
E.g. ./mtg1.3 -p test.fam -d test.dat -g test.grm -cc test.cov -qc test.pc -out test.out -sv test.sv
This is optional, but sometime you need a set of good starting values for a proper convergence (they may come from previous univariate analyses). The format of test.sv file should be as follows

(for 5 traits model with one random effects model)

```
ve 0.11  #first trait residual
ve 0.14  #second trait residual
ve 0.21  #third trait residual
ve 0.24  #fourth trait residual
ve 0.14  #fifth trait residual

va 0.13    #first trait genetic variance (g1)
va 0.15    #second trait genetic variance (g2)
va 0.23    #first trait genetic variance (g3)
va 0.11    #second trait genetic variance (g4)
va 0.19    #second trait genetic variance (g5)

cov 0.04  #covariance between g1,g2
cov 0.02  #covariance between g1,g3
cov 0.05  #covariance between g2,g3
cov 0.01  #covariance between g1,g4
cov 0.02  #covariance between g2,g4
cov 0.00  #covariance between g3,g4
cov 0.07  #covariance between g1,g5
cov 0.04  #covariance between g2,g5
cov 0.08  #covariance between g3,g5
cov 0.02  #covariance between g4,g5
```

When modelling residual covariance (i.e. -cove 1)
E.g. ./mtg1.3 -p test.fam -d test.dat -g test.grm -cc test.cov -qc test.pc -out test.out -cove 1 -sv test.sv

```
ve 0.11  #first trait residual (e1)
ve 0.14  #second trait residual (e2)
ve 0.21  #third trait residual (e3)
ve 0.24  #fourth trait residual (e4)
ve 0.14  #fifth trait residual (e5)

cov 0.01  #covariance between e1,e2
cov 0.02  #covariance between e1,e3
cov 0.03  #covariance between e2,e3
cov 0.02  #covariance between e1,e4
cov 0.01  #covariance between e2,e4
cov 0.00  #covariance between e3,e4
cov 0.06  #covariance between e1,e5
cov 0.01  #covariance between e2,e5
cov 0.00  #covariance between e3,e5

va 0.13    #first trait genetic variance (g1)
va 0.15    #second trait genetic variance (g2)
va 0.23    #first trait genetic variance (g3)
va 0.11    #second trait genetic variance (g4)
va 0.19    #second trait genetic variance (g5)

cov 0.04  #covariance between g1,g2
cov 0.02  #covariance between g1,g3
cov 0.05  #covariance between g2,g3
cov 0.01  #covariance between g1,g4
cov 0.02  #covariance between g2,g4
cov 0.00  #covariance between g3,g4
cov 0.07  #covariance between g1,g5
cov 0.04  #covariance between g2,g5
cov 0.08  #covariance between g3,g5
```

-mg {file name} instead of -g {file name}: multiple random effects model
E.g. ./mtg1.3 -p test.fam -d test.dat -mg grm_list.txt -cc test.cov -qc test.pc -out test.out -mod 5

The file which here is named grms_list.txt should be a text file, containing the file names of each grm file,
test.grm1
test.grm2
……

-mbg {file name} instead of -bg {file name}: multiple random effects model with binary GRM format
E.g. ./mtg1.3 -p test.fam -d test.dat -mbg grm_list.txt -cc test.cov -qc test.pc -out test.out -mod 5

The file which here is named grms_list.txt should be a text file, containing the file names of each binary grm file,
test.grm1.bin
test.grm2.bin
……

-bv {file name}: BLUP estimation
E.g. ./mtg1.3 -p test.fam -d test.dat -g test.grm -cc test.cov -qc test.pc -out test.out -mod 5 -bv test.bv
Then this will ouput test.bv and test.bv.py

*.bv contains predicted risk scores for each trait (the first line is for fixed effects solutions).
*.bv.py contains the terms in the right hand side except the scaled and standardised SNP coefficients in eq. (3) , i.e.

$$\begin{bmatrix} \sigma_{g_1}^2 & \cdots & \sigma_{g_{1n}} \\ \vdots & \ddots & \vdots \\ \sigma_{g_{n1}} & \cdots & \sigma_{g_n}^2 \end{bmatrix} \otimes \mathbf{I} \cdot \begin{bmatrix} \mathbf{Z}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{Z}_n \end{bmatrix}' \cdot \mathbf{V}^{-1} \begin{bmatrix} \mathbf{y}_1 - \mathbf{X}_1 b_1 \\ \vdots \\ \mathbf{y}_n - \mathbf{X}_n b_n \end{bmatrix} M^{-1}$$

This will be used to get SNP BLUP in the multivariate framework (see the next section). GCTA has already used this intermediate variable for the univariate model.

3. Converting individual BLUP to SNP BLUP in multivariate frame work

In order to convert individual BLUP to SNP BLUP, we introduce another computer program named ‘snp_blup1.5’.

<In discovery set>

```
./snp_blup1.5 {plink bed file prefix} {a part from *.bv.py} {option a or b} {output file name}
```

For example, If you want to get SNP BLUP for the trait t (tth trait), do the following with the BLUP output file from mtg1.3 (e.g. test.bv.py).

```
awk ‘$1==t {print $2}’ test.bv.py > tmp
./snp_blup1.5 test tmp a test.snpv
```

The output file test.snpv has the same format as --score in PLINK, e.g.

```
   SNP ID, Reference allele, Score (numeric)
```

for example

```
   SNPA   A    1.95
   SNPB   C    2.04
   SNPC   C   -0.98
   SNPD   C   -0.24
```

<In validation set>

If you want to get individual BLUP for the trait t for the validation set, do the following.

```
./snp_blup1.5 valid test.snpv b valid.gbv
```

The program with the option b detects reference alleles in the discovery set that are not matched to that in the validation set, e.g.

```
reference matched             :       2556
reference filp SNP # (fliped) :          0
strand filp SNP # (fliped)    :          0
strand flip + reference flip  :          0
ambiguous SNPs                :        321
```

It would be recommended to exclude any unmatched reference alleles or ambiguous alleles before running the program with the b option.

When using snp_blup1.5, it implicitly reads a file having
Chromosome number, SNP ID, 2 x allele frequency (i.e. mean(x), variance of SNP coefficient (i.e. var(x))

A user can provide this information or ‘rtmx_frq-vx’ (put in the .zip file) can be used.

./rtmx_frq-vx {plink bed file prefix}: this will give 2 x reference allele frequency and var(x) where x is 0, 1, 2 SNP coefficients.
e.g. ./rtmx_frq-vx test
This will generate test.freq.

Or,

```
./rtmx_frq-vx valid
```

This will generate valid.freq.

Reference

1. Yang, J., Lee, S.H., Goddard, M.E. & Visscher, P.M. GCTA: A tool for genome-wide complex trait analysis. *Am J Hum Genet* **88**, 76-82 (2011).
2. Lee, S.H. & Van der Werf, J.H.J. An efficient variance component approach implementing an average information REML suitable for combined LD and linkage mapping with a general complex pedigree. *Genet Sel Evol* **38**, 25-43 (2006).
3. Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A.R., Bender, D., Maller, J., Sklar, P., de Bakker, P.I.W., Daly, M.J. *et al.* PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am J Hum Genet* **81**, 559-575 (2007).