CREATION AND HANDLING OF GENOMIC RELATIONSHIP MATRICES WITH PREGSF90

I. Aguilar

Instituto Nacional de Investigación Agropecuaria
INIA Las Brujas, Uruguay

Genomic Relationship Matrix - G

- G = ZZ'/k
  - Z = matrix for SNP marker
  - Dimension Z= n*p
  - n animals,
  - p markers

Data file with SNP marker

```
80    2110101100201201101101011011111121111121010 0
8014  2111010151110112022111011151111210111221010 0
516   2110010120225202112021012110211120221211110 1
181   2111011111220112055020002010102221221111110 0
```

HOWTO: Creation of Genomic Matrix

- Read SNP marker information => M $\begin{bmatrix} 2 & 1 & 2 & .. \\ 0 & 1 & 0 & .. \\ .. & .. & .. & .. \end{bmatrix}$
- Get 'means' to center
  - Calculate allele frequency from observed genotypes (p_i)
  - p_i= sum($SNPcode_i$)/2n
- Matrix for center W(3,p) $\begin{matrix} 0 \\ 1 \\ 2 \end{matrix} \begin{bmatrix} 0-2p_1 & 0-2p_2 & .. \\ 1-2p_1 & 1-2p_2 & .. \\ 2-2p_1 & 2-2p_2 & . \end{bmatrix}$
- Center matrix Z = W(M)

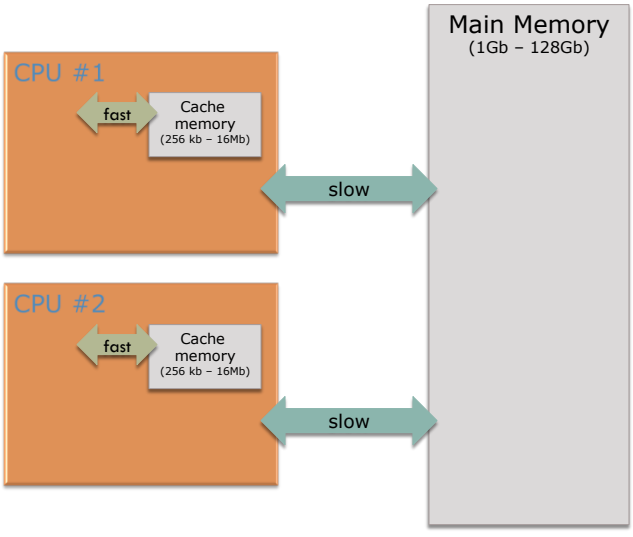
Creation of Genomic

- Issues
  - Large number of genotyped individuals
  - Large number of SNP markers
  - Matrix multiplication ~ cost n^2 * p
- Large amount of data put in (cache) memory for doing 'matmul' for each pair of animals and indirect memory access (center)
  - Memory hierarchy

Matrix multiplication

- Matrix multiplication
  - Several methods
    - Intrisic matmul (good for small examples !!!)
    - "do-loops"
    - Packages (BLAS, LAPACK)
      - Non-optimzed
      - Optimized (ATLAS, MKL, etc.)
  - Several Compilers
    - Perform automatic optimization
      - Vectorize loops
      - Detect permuted loops
    - Can use OpenMP directives for parallelization

Memory Hierarchy



Alternative codes to create G matrix

Original

```
Do i=1,n
      Do j=i,n
            S=0
            Do k=1,p
                  S=S+Z(M(i,k),k)
                    *Z(M(j,k),k)
            End do
            G(i,j)=S/sqrt(d(i)*d(j))
            G(j,i)=G(i,j)
      End do
End do
```

Optimize Indirect Memory Access -OPTM

```
Do k=1,p
      X(:,k)=Z(M(:,k),k)
End do
Do i=1,n
      Do j=i,n
            S=0
            Do k=1,p
                  S=S+X(i,k)
                    *X(j,k)
            End do
            G(i,j)=S/sqrt(d(i)*d(j))
            G(j,i)=G(i,j)
      End do
End do
```

Optimize Memory and Loops - OPTML

```
Do k=1,p
      X(:,k)=Z(M(:,k),k)
End do
Do i=1,n
      Do j=1,n
            Do k=1,p
                  G(i,j)=G(i,j)
                    +X(i,k)*X(j,k)
            End do
      End do
End do
Do i=1,n
      Do j=1,n
            G(i,j)=G(i,j)/sqrt(d(i)*d(j)
      End do
End do
```

Gmatrix.f90 (VanRaden, 2009)

CPU time for alternative codes for G matrix and machines

- Testing
  - 6500 genotyped animals
  - 40k SNPs

		Algorithms		
Processor	**Cache**	**Original**	**OPTM**	**OPTML**
Xeon 3.5 GHz	6 MB	24 m	26 m	7 m
Opteron 3.0 GHz	1 MB	265 m	59 m	17 m

CPU time (m) with alternative codes and compilers

- Testing
  - 6500 genotyped animals
  - 40k SNPs
  - Opteron 3.02 GHz 1 MB Cache memory

Compiler	Original	OPTM	OPTML
Intel	265	59	17
Absoft	241	60	34
gfortran	213	63	>1day

`PreGSf90` program

- From BLUPF90 package
- Uses a genomic module
- Creation and handling of genomic relationship matrices and relationship based on pedigree
- Different methods to optimize calculations using parallel processing

Input files

- Same parameter file as for all BLUPf90 programs
  - But with "OPTION SNP_file xxxx"
  - indicate to run genomic subroutines
- Pedigree file
- Marker information (SNP file)
- Cross Reference file for renumber ID
  - Links genotypes files with codes in pedigree, etc.

OPTIONS – BLUPF90 parameter file

- PreGSF90
  - controled by adding OPTIONS commands to the parameter file
  - `OPTION SNP_file` *`marker.geno.clean`*
  - Read 2 files:
    - `marker.geno.clean`
    - `marker.geno.clean.XrefID`

RENUMF90

- Add keyword to the "animal effect"
  `SNP_FILE`
  `marker_geno_clean`
- Renumber tool to prepares:
  - data
  - pedigree
  - genotypes
  - parameter files for BLUPF90 programs including PREGSF90
- Check wiki:
- http://nce.ads.uga.edu/wiki/doku.php

Parameters file

RENUMF90
renum.par

```
DATAFILE
phenotypes.txt
TRAITS
3
FIELDS_PASSED TO OUTPUT

WEIGHT(S)

RESIDUAL_VARIANCE
 0.9038
EFFECT
1 cross alpha  # mu
EFFECT
2 cross alpha  # animal
RANDOM
animal
FILE
pedigree
SNP_FILE
marker.geno.clean
(CO)VARIANCES
    0.9951E-01
```

BLUPF90
renf90.par

```
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
          1
NUMBER_OF_EFFECTS
          2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBE
  2         1 cross
  3     15800 cross
RANDOM_RESIDUAL VALUES
  0.90380
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped
(CO)VARIANCES
  0.99510E-01
OPTION SNP_file marker.geno.clean
```

Pedigree file from RENUMF90

- 1 - **animal number**
- 2 - parent 1 number or UPG
- 3 - parent 2 number or UPG
- 4 - 3 minus number of known parents
- 5 - known or estimated year of birth
- **6** - number of known parents;
  if animal is genotyped 10 + number of known parents
- 7 - number of records
- 8 - number of progenies as parent 1
- 9 - number of progenies as parent 2
- **10 - original animal ID**

SNP file & Cross Reference Id

SNP File First col: Identification, could be alphanumeric
Second col: SNP markers {codes: 0,1,2 and 5 for missing}

```
80   2110101100201201101101011011111
8014 2111010151110112022111011151111
516  2110010120225202112021012110 21
181  2111011111220112055020002010 10
```

Renumber ID

Cross Reference ID

```
1732 80
8474 8014
406 516
9441 181
```

Pedigree File (from RENUMF90)

```
1732 11010 10584 1 3 12  1 0 0 80
8474  8691  9908 1 3 12  1 0 0 8014
 406  8691  9825 1 3 12  1 0 2 516
9441  8691  8829 1 3 12  1 0 0 181
```

Original ID

Genomic Matrix default options

- G* = ZZ'/k as in VanRaden, 2008
- With:
  - Z center using allele frequencies estimated from the genotyped individuals
  - k = 2 sum (p * (1-p))
- G = G*0.95 + A*0.05 (to invert)
- Tunning of G (see Z. Vitezica talk)
  - Adjust G to have mean of diagonals and off-diagonals equal to A

Genomic Matrix Options

- OPTION whichG x
  - 1: G=ZZ'/k (default) (VanRaden, 2008)
  - 2: G=ZDZ'/n; D=1/2p(1-p) (Amin et al., 2007; Leuttenger et al., 2003)
  - 3: As 2 with modification UAR (Yang et al., 2010)
- OPTION weightedG *file*
  - Read weights to create G=ZDZ'
  - Weighting Z*= Z sqrt(D) => G = Z*Z*' = ZDZ'
- OPTION whichScale x
  - 1: 2 Σ (p(1-p)) (default) (VanRaden, 2008)
  - 2: trace(ZZ')/n (Legarra 2009, Hayes 2009, Forni et al 2011)
  - 3: correction (Gianola et al., 2009)

Genomic Matrix Options

- OPTION whichfreq x
  - 0: read from file *freqdata* or other specified
  - 1: 0.5
  - 2: current calculated from genotypes (default)
- OPTION FreqFile *file*
  - Reads allele frequencies from a file
- OPTION maxsnps x
  - Set the maximum length of string for reading marker data from file => BovineHD chip

Options for Blending G and A

- OPTION AlphaBeta alpha beta
  - $G = alpha*G^r + beta*A$
- OPTION tunedG
  - 0: no adjustment
  - 1: mean(diag(G))=1, mean(offdiag(G))=0
  - 2: mean(diag(G))=mean(diag(A)), mean(offdiag(G))=mean(offdiag(A)) (default)
  - 3: mean(G)=mean(A)
  - 4: Use Fst adjustment. Powell et al. (2010) & Vitezica et al. (2011)

Creation of 'raw' genomic matrix

- Tricks:
- Use dummy pedigree

  1 0 0
  2 0 0
  …
- Change blending parameters
  - OPTION AlphaBeta 0.99 0.01
- No adjustment for compatibility with A
  - OPTION tunedG 0

$$G = 0.99*G + 0.01*I$$

Storing and Reading Matrices

- PreGSF90:
  - Facilitate the implementation of single-step, (tomorrow)
  - Matrix A is replaced by H with:

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

  - Default output is the matrix GimA22i, to be included in apllication programs (BLUPF90, REMLF90..)
- BUT: intermediate matrices could be stored for examination, use in application programs, etc.

Storing and Reading Matrices

- Matrices that can be stored:
  - A22, inv(A22), G, inv(G), GmA22, inv(GmA22), inv(H)
- All matrices are stored in same format:
  - upper triangle
  - By default in binary format
  - But to store in text (Ascii) format:
    - Use: `OPTION saveAscii`
- Values
  - i j val
  - i & j refers to the row number in the genotype file !!!!!
  - Renumber ID could be obtained from the XrefID file

Storing and Reading Matrices

To save our 'raw' genomic matrix:

- OPTION saveG [all]
  - If the optional *all* is present all intermediate G matrices will be saved!!!

or it inverse

- OPTION saveGInverse
  - Only the final matrix G, after blending, scaling, etc. is inverted !!!

- Look in wiki for keywords for other matrices

Storing with Original IDs

- Some matrices could be stored in text files with the original IDs extracted from *renaddxx.ped* created by the RENUMF90 program (col #10)
- For example:
  - OPTION saveGOrig
  - OPTION saveDiagGOrig
  - OPTION saveHinvOrig
- Values
  - origID_i, origID_j, val

OUTPUT

- Only GimA22i , other requested matrices files, and some reports (tomorrow) are stored.
- Main log is printout to the screen !!!
- Use redirection '>'
- or better the command `tee` to save in a log file.
- This will allows to save and see the messages from the program
- `echo renf90.par | preGSf90 | tee pregs.log`

Printout: Same heading as other programs

```
Options read from parameter file:

 * SNP file: marker.geno.clean
 * SNP Xref file: marker.geno.clean_XrefID
 * Matrix in Ascii format(default=binary)

 *-----------------------------------------------------------------*
 *                  Genomic Library: Version 1.110                 *
 *                                                                 *
 *  Modified relationship matrix (H) created for effect:   2       *
 *-----------------------------------------------------------------*

 Read 18600 animals from pedigree file: "renadd02.ped"
 Number of Genotyped Animals: 1500
```

All options that were enter in the parameter file should be here !!. IF not check that keywords are correct (upper and lower case)

Check number of animals and individuals with genotypes

Printout

```
Creating A22
   Extracting subset of: 4634 pedigrees from: 18600 elapsed time:     0.0019
   Calculating A22 Matrix by Colleau ...elapsed time   1.250464

Reading SNP file
   Column position in file for the first marker: 7
   Format to read SNP file: (6x,400000i1)
   Number of SNPs: 3000
   Number of Genotyped animals: 1500
   Reading SNP file elapsed time: .41

Statistics of alleles frequencies in the
   N:           3000
   Mean:       0.500
   Min:        0.101
   Max:        0.898
   Var:        0.016
```

Information from genotype file. The format is detected from the first line !!!

So all genotypes should start in the same column !!!

Number of SNP is also determined by the first line!!

Looking stored matrices

- Avoid open with text editors, huge files !!!
- For example:
- 1500 genotyped individuals => 1,125,750 rows
- Inspection could be done by Unix commands:
  - `head G` => first 10 lines
  - `tail G` => last 10 lines
  - `less G` => scroll document by line/page
  - `wc -l G` => count number of lines
    good for checks with the number of genotypes (n) = (n*(n+1)/2)

head G

```
1 1 .999382118619
1 2 .355052761478
2 2 1.014521277458
1 3 -.048184197960
2 3 -.057513012886
3 3 .976558921904
1 4 -.101734083083
2 4 -.007644724611
3 4 .196757165096
4 4 1.018165021903
```

GBLUP, GREML, GGIBBS

- Using BLUPF90 programs to perfom genomic selection using genomic relationship matrix
- Using only phenotypes or pseudo phenotypes (DYD, DP, EBV) for only genotyped individuals

Two ways: user_file

- By user defined files for covariances of random effects
- Look at Tricks in the wiki for more details http://nce.ads.uga.edu/wiki/doku.php
- Special type of random effect in BLUPF90 parameter file
- Gi created by PreGSF90 can be used here!

```
RANDOM_GROUP
 # genomic
 2
RANDOM_TYPE
user_file
FILE
 # matrix file
Gi
```

By 'fake' single-step GBLUP

- Same trick as before:
  - Dummy pedigree with number of individual equal to number of individuals with genotypes
  - Little blending with A (identity matrix) to create the inverse (OPTION AlphaBeta 0.99 0.01)
  - No adjustment for means of A (OPTION tunedG 0)
  - Parameter file include:
    - Random effect defined as **add_animal**
    - OPTION SNP_file xxxx

By 'fake' single-step GBLUP

- Runs could be either by:
  - Several steps
    - 1 run **preGSf90** and store G inverse
    - 2 modify paramter file for BLUP

      adding OPTION readGimA22i
    - 3 run **BLUPF90**
  - 'One-Step'
    - 1 run **BLUPF90 or REMLF90**

RENUMF90 ren.par

```
DATAFILE
mice_records.txt
TRAITS
1
FIELDS_PASSED TO OUTPUT

WEIGHT(S)

RESIDUAL_VARIANCE
 0.16
EFFECT
6 cross alpha  # sex
EFFECT
5 cross alpha  # animal
RANDOM
animal
FILE
ped_unrelated
SNP_FILE
mice_genotypes.txt
(CO)VARIANCES
    4.59
EFFECT
7 cross alpha  # cage
RANDOM
diagonal
(CO)VARIANCES
    2.12
OPTION tunedG 0
OPTION AlphaBeta 0.99 0.01
```

BLUPF90 renf90.par

```
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
          1
NUMBER_OF_EFFECTS
          3
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVEL
S TYPE_OF_EFFECT[EFFECT NESTED]
  2         2 cross
  3      1884 cross
  4       531 cross
RANDOM_RESIDUAL VALUES
  0.16000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped
(CO)VARIANCES
   4.5900
 RANDOM_GROUP
     3
 RANDOM_TYPE
 diagonal
 FILE

(CO)VARIANCES
   2.1200
OPTION SNP_file mice_genotypes.txt
OPTION tunedG 0
OPTION AlphaBeta 0.99 0.01
```

PreGSf90 inside BLUPF90 ??

- Almost all programs from package support creation of genomic relationship matrices, Hinv, etc.
- OPTION SNP_file xxxx

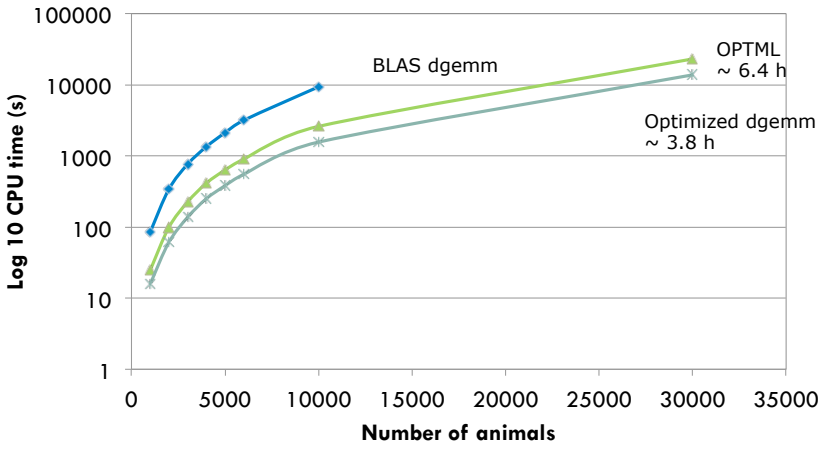
- Why preGSF90 ?
  - Same genomic relationship matrix for several models, traits, etc. Just do it once and store.
  - Uses of optimized subroutines for efficient matrix multiplications, inversion and with support for parallel processing

Matrix multiplication subroutines

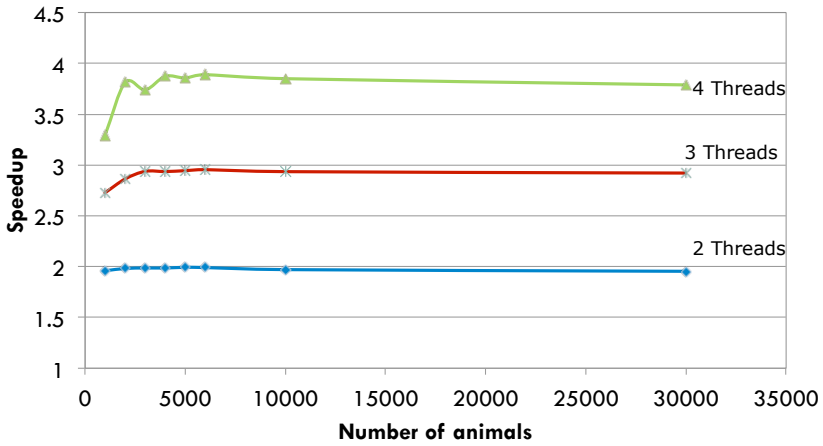
- Optimized memory and loops (compiler optimization)
- *dgemm* subroutine from BLAS
- Optimized *dgemm* (ATLAS or MKL libraries*)
  - Serial
  - Parallel (Automatic use of OpenMP)

* Intel Fortran Compiler

Matrix multiplication using 40k SNPs



Speedup for matrix multiplications



Speedup = time using one thread/time using *n* threads

Computing time with 4 processors

Number of genotypes	Creation of G	Inversion
10k	2 m	2 m
30k	1 h	1 h
50k	2.5 h	4.5 h

Creation a subset of relationship matrix (A22)

- Create a relationship matrix for only genotyped animals (~ thousands)
- Full pedigree (~millions)
- Trace only ancestors of genotyped (reduce but still large number for A matrix)

Relationship Matrix of Genotyped Animals

- Colleau's algorithm to creates A_{22}
- No need to have explicit A matrix
- Method uses "matrix-vector" multiplication with a decomposition of A matrix

$$\mathbf{v} = \mathbf{Ar} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{P})^{-1'}\mathbf{r}$$

Example A times a vector

Pedigree

	[,1]	[,2]	[,3]
[1,]	1	0	0
[2,]	2	0	0
[3,]	3	1	2

Matrix P

	[,1]	[,2]	[,3]
[1,]	0.0	0.0	0.0
[2,]	0.0	0.0	0.0
[3,]	0.5	0.5	0.0

Matrix $(I-P)^{-1}$

	[,1]	[,2]	[,3]
[1,]	1.0		
[2,]	0.0	1.0	
[3,]	0.5	0.5	1.0

$$\mathbf{v} = \mathbf{Ar} = \mathbf{(I - P)^{-1}D(I - P)^{-1'}r}$$

Matrix $(I-P)^{-1}$

	[,1]	[,2]	[,3]
[1,]	1.0		
[2,]	0.0	1.0	
[3,]	0.5	0.5	1

Matrix D

	[,1]	[,2]	[,3]
[1,]	1		
[2,]		1	
[3,]			0.5

Vector q

	[,1]
[1,]	25
[2,]	35
[3,]	30

=

Matrix $(I-P)^{-1'}$

	[,1]	[,2]	[,3]
[1,]	1	0	0.5
[2,]		1	0.5
[3,]			1.0

Vector r_2

	[,1]
[1,]	10
[2,]	20
[3,]	30

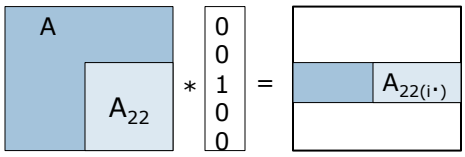
```
Do i=1,n
      vi = qi*di+(qsi+qdi)/2
End do
```

```
Do i=n,1
      qi = qi+r2i
      qsi = qsi+qi/2
      qdi = qdi+qi/2
End do
```

Relationship Matrix of Genotyped Animals

- For each genotyped animal in A_{22}

$$\mathbf{v} = \mathbf{Ar_2} = \mathbf{(I - P)^{-1}D(I - P)^{-1'}r_2}$$



Tabular method vs. Colleau algorithm

- Testing
  - 6,500 genotyped Holsteins
  - 57,000 pedigrees

	Tabular*	Colleau method
CPU Time	311 s	45 s
Memory	12.1GB	322MB

* Gmatrix.f90 (VanRaden, 2009)