Lab3 Quality control of SNP data and GBLUP

Prepared by D. Lourenco, I. Aguilar, A. Legarra, and Z. Vitezica

The data for this lab is based on a public pig dataset from PIC (Cleveland et al. 2012 - G3 Journal). Originally, this dataset was filtered for MAF and missing SNP were imputed, however some modifications were introduced to generate commons problems that are found in real datasets.

Files are available in the folder:
`/data1/RFN201908/daniela/labs/lab3`

Description of files

phenotypes_new.txt:

1: Animal ID
2: Trait 1
3: Trait 2
4: Trait 3
5: Trait 4
6: Trait 5
7: Mean

pedigree_new.txt :

1: animal ID
2: sire ID
3: dam ID

genotypes_new.txt :

1: animal ID
2: marker information

1. Copy the full folder into your directory
   `cp –r /data1/RFN201908/daniela/labs/lab3 .`

2. Using Unix commands, create a dummy pedigree file and a file with phenotypes only for genotyped animals. Those files will be used in GBLUP.

3. Run renumf90 using PED_DEPTH 1

4. Run preGSf90 to get statistics for the SNP data
   - Check the initial number of SNPs, all statistics related to SNPs, reasons why SNPs did not pass the quality control.

   Remember that preGSf90 does the quality control and sets up the genomic and pedigree relationship matrices for genotyped animals. To avoid the matrix construction and perform only quality control, use the following options:

   ```
   OPTION createG 0
   OPTION createGInverse 0
   OPTION createA22 0
   OPTION createA22Inverse 0
   OPTION createGimA22i 0
   ```

5. Run preGSf90 WITH quality control again saving the clean files at this time. Check the number of animals and number of SNP in the clean files. Avoid the construction of genomic and pedigree relationship matrices for genotyped animals.

6. With the clean genotype file, run GBLUP in blupf90. Check the options you need to include in the parameter file:
   http://nce.ads.uga.edu/wiki/doku.php?id=how_to_run_pure_gblup
   Check the output of blupf90 and the solution file.

7. Let's assume you are working on a project and your objective is to test different models using the same data. You can run preGSf90 with clean data once and save **G**. Everytime you change your model, you can just read **G** from a file avoiding the creation of this matrix everytime. This can save some computing resources. Check the documentation for preGSf90 and explore the options to save **G**.
   http://nce.ads.uga.edu/wiki/doku.php?id=readme.pregsf90

   `OPTION saveG`
   Run blupf90 with an option to read **G**. Compare the current solutions with solutions from exercise 6.

8. blupf90 has an interesting option where an external covariance matrix can be used. This is especially useful when different relationship matrices are needed (e.g., polyploidy populations) or dominance effects are to be considered. Check how this can be done:
   http://nce.ads.uga.edu/wiki/doku.php?id=user_defined_files_for_covariances_of_random_effects

   Run preGSf90 with an option to save $\mathbf{G}^{-1}$ in text format
   `OPTION saveAscii` and `OPTION saveGInverse`

   Run blupf90 with the option to read an external covariance matrix.
   Be aware that the first two columns in $\mathbf{G}^{-1}$ are the position of genotyped animals in the genotype and genotype_XrefID files. When you use an user file in blupf90, IDs in the covariance matrix should match IDs in the phenotype file.

   Before running blupf90, you can change the IDs for the animals in the phenotype file using the following commands:

```
awk '{print $1,NR}' genotypes.txt_clean_XrefID |
sort +0 -1 > index.gen
awk '{print $3,$0}' renf90.dat | sort +0 -1 >
srenf90.temp
join -1 +1 -2 +1 srenf90.temp index.gen | awk
'{print $2,$3,$5,$4}' | sort -n +2 -3 > srenf90.dat
```

   Do not forget the IDs in solutions are now the position of genotyped animals in the genotype and genotype_XrefID files!