

Lab2 Quality control of SNP data, GBLUP, and ssGBLUP

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 common problems that are found in real datasets.

Files are available in the website:

Use curl to download it to your Linux or Mac device:

```
curl http://nce.ads.uga.edu/wiki/lib/exe/fetch.php?media=lab1_UF.zip -o lab2.zip
```

Description of files

phenotypes.txt:

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

pedigree.txt :

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

genotypes.txt :

- 1: animal ID
- 2: marker information

1. Look into the files and identify if all individuals in the pedigree have genotypes
2. Run `renumf90` in the single-trait context.
3. Run `preGSf90` to get statistics for the SNP data, saving the clean files
- 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 createGInverse 0
OPTION createA22Inverse 0
OPTION createGimA22i 0
```

4. Run `ssGBLUP` using the clean SNP file. Don't forget to turn QC off. If you want to get SE of GEBV, include `OPTION sol se`. Check the output of `blupf90` and the solution file (`solutions`)
5. Trick `BLUPF90` to run GBLUP: Copy `phenotypes.txt`, `pedigree.txt`, and `genotypes.txt_clean` to a separate folder. Using Unix commands, create a dummy pedigree file and a file with phenotypes only for genotyped animals.
6. Run `renumf90` using `PED_DEPTH 1`

7. Run GBLUP in blupf90. Check the options you need to include in the parameter file (slides day2_3). Check the output of blupf90 and solutions
8. 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**. Every time you change your model, you can just read **G** from a file avoiding the creation of this matrix every time. 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>

Run preGSf90 and save **G**. Run blupf90 with an option to read **G**. Compare the current solutions with solutions from exercise 8.

9. 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_r_andom_effects

Run preGSf90 with an option to save **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 **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!