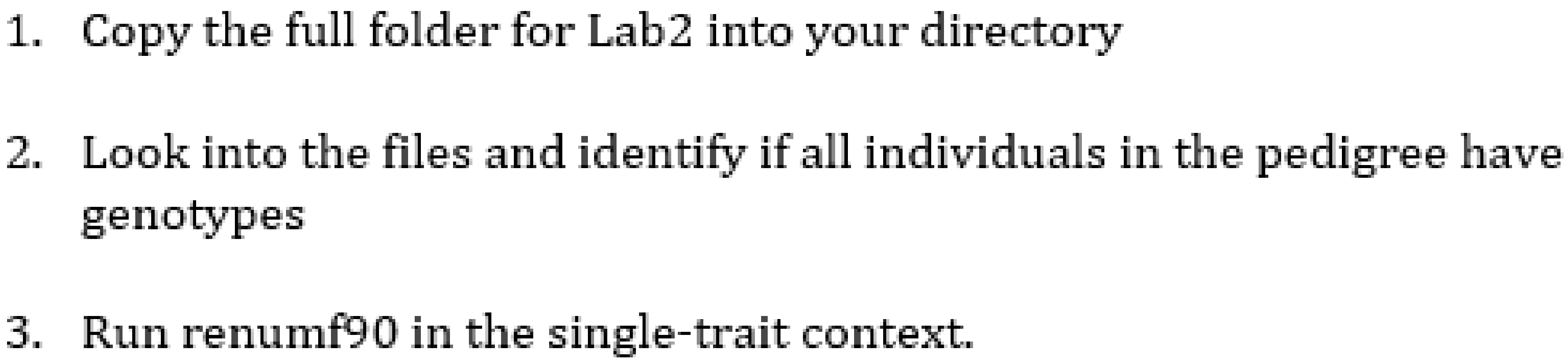
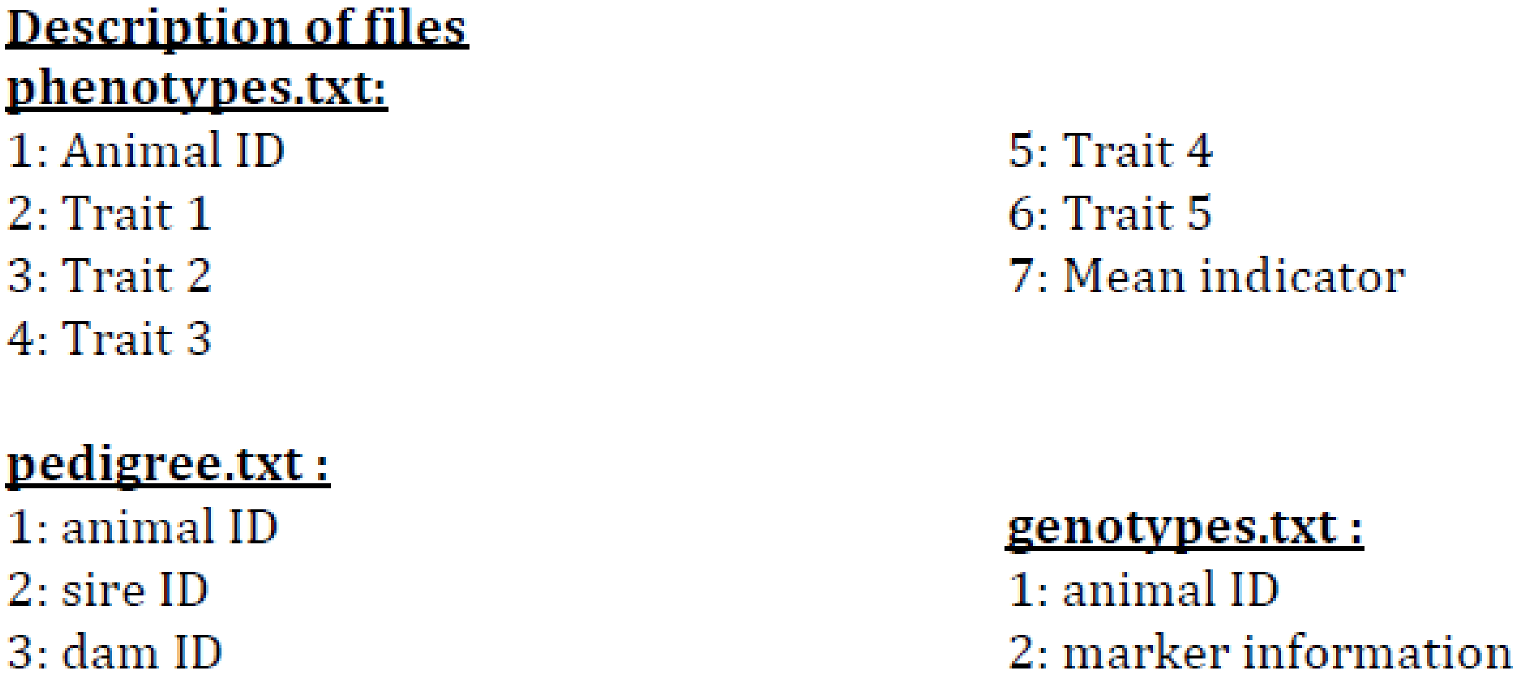
Block 1



1. Copy the full folder for Lab2 into your directory
2. Look into the files and identify if all individuals in the pedigree have genotypes
3. Run renumf90 in the single-trait context.

The dataset



Description of files

phenotypes.txt:

1: Animal ID

2: Trait 1

3: Trait 2

4: Trait 3

5: Trait 4

6: Trait 5

7: Mean indicator

pedigree.txt :

1: animal ID

2: sire ID

3: dam ID

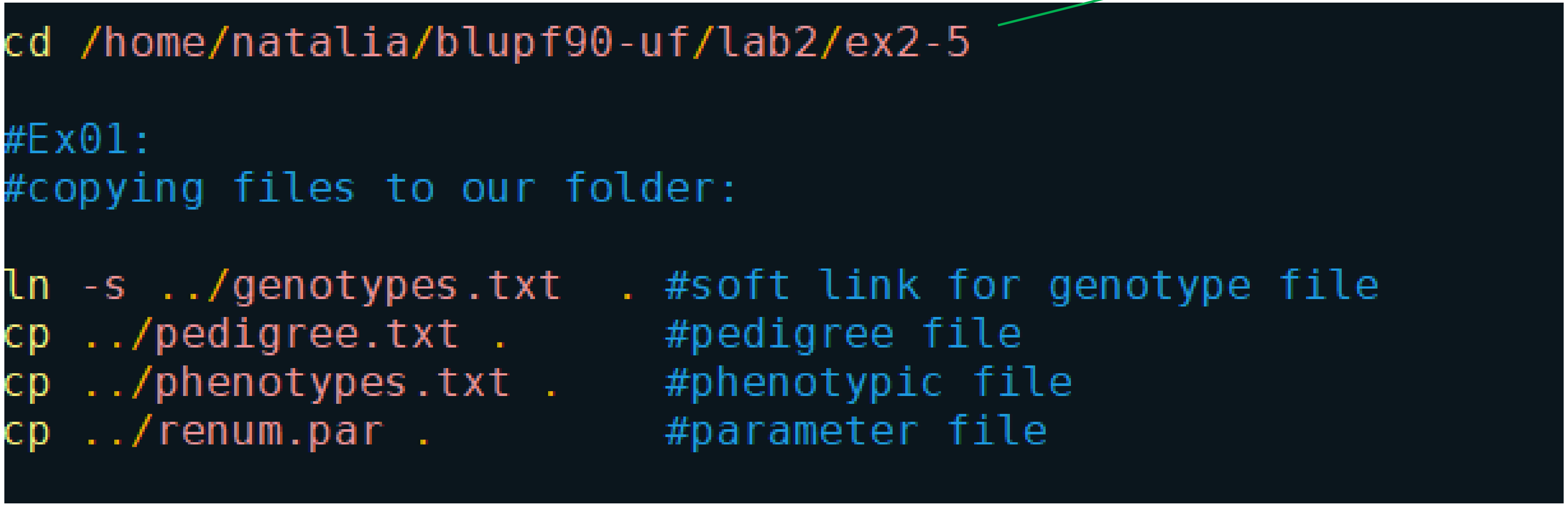
genotypes.txt :

1: animal ID

2: marker information

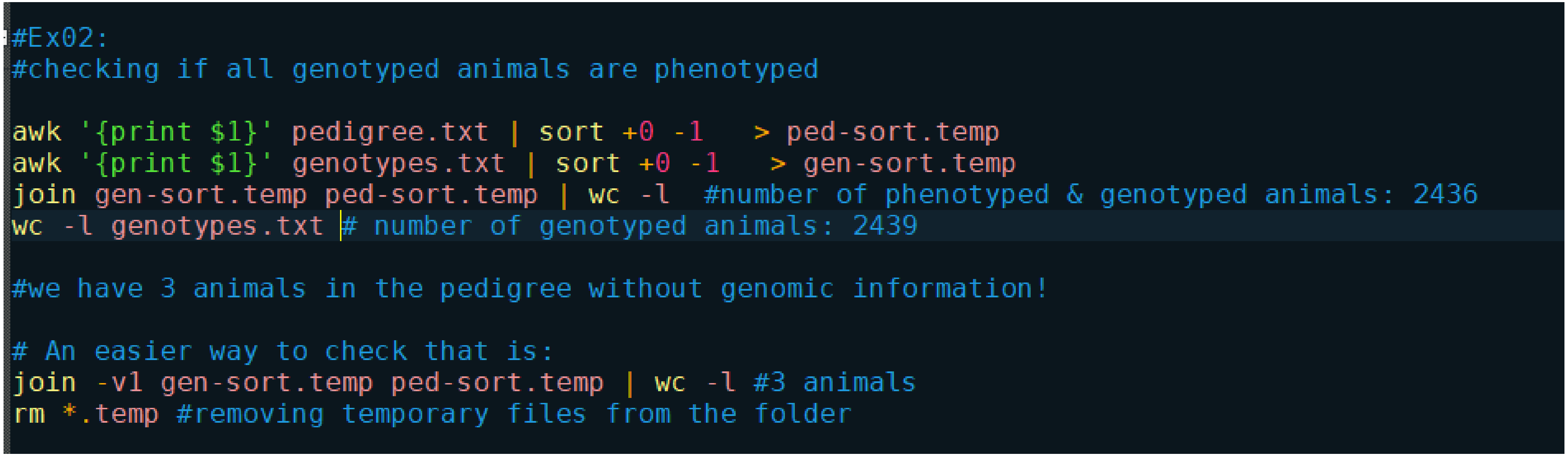
1. Copy the full folder for Lab2 into your directory

My folder address



```
cd /home/natalia/blupf90-uf/lab2/ex2-5

#Ex01:
#copying files to our folder:

ln -s ../genotypes.txt  . #soft link for genotype file
cp ../pedigree.txt .      #pedigree file
cp ../phenotypes.txt .    #phenotypic file
cp ../renum.par .         #parameter file
```

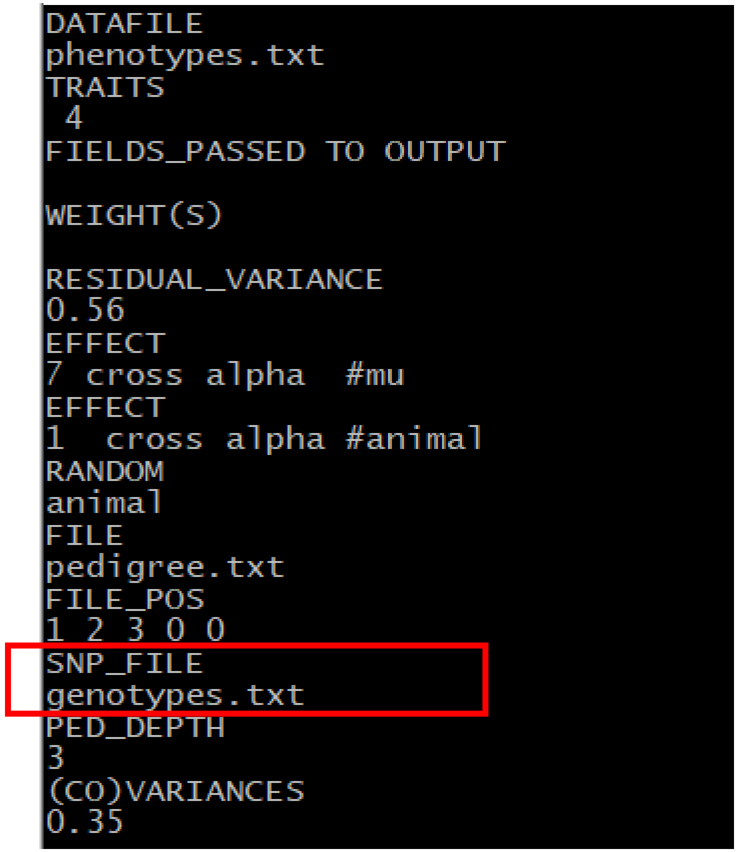
2. Look into the files and identify if all individuals in the pedigree have genotypes



```
#Ex02:
#checking if all genotyped animals are phenotyped

awk '{print $1}' pedigree.txt | sort +0 -1   > ped-sort.temp
awk '{print $1}' genotypes.txt | sort +0 -1   > gen-sort.temp
join gen-sort.temp ped-sort.temp | wc -l  #number of phenotyped & genotyped animals: 2436
wc -l genotypes.txt # number of genotyped animals: 2439

#we have 3 animals in the pedigree without genomic information!

# An easier way to check that is:
join -v1 gen-sort.temp ped-sort.temp | wc -l #3 animals
rm *.temp #removing temporary files from the folder
```

3. Run renumf90 in the single-trait context.

renum.par



```
DATAFILE
phenotypes.txt
TRAITS
 4
FIELDS_PASSED TO OUTPUT

WEIGHT(S)

RESIDUAL_VARIANCE
0.56
EFFECT
7 cross alpha  #mu
EFFECT
1  cross alpha #animal
RANDOM
animal
FILE
pedigree.txt
FILE_POS
1 2 3 0 0
SNP_FILE
genotypes.txt
PED_DEPTH
3
(CO)VARIANCES
0.35
```

Description of files

phenotypes.txt:

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

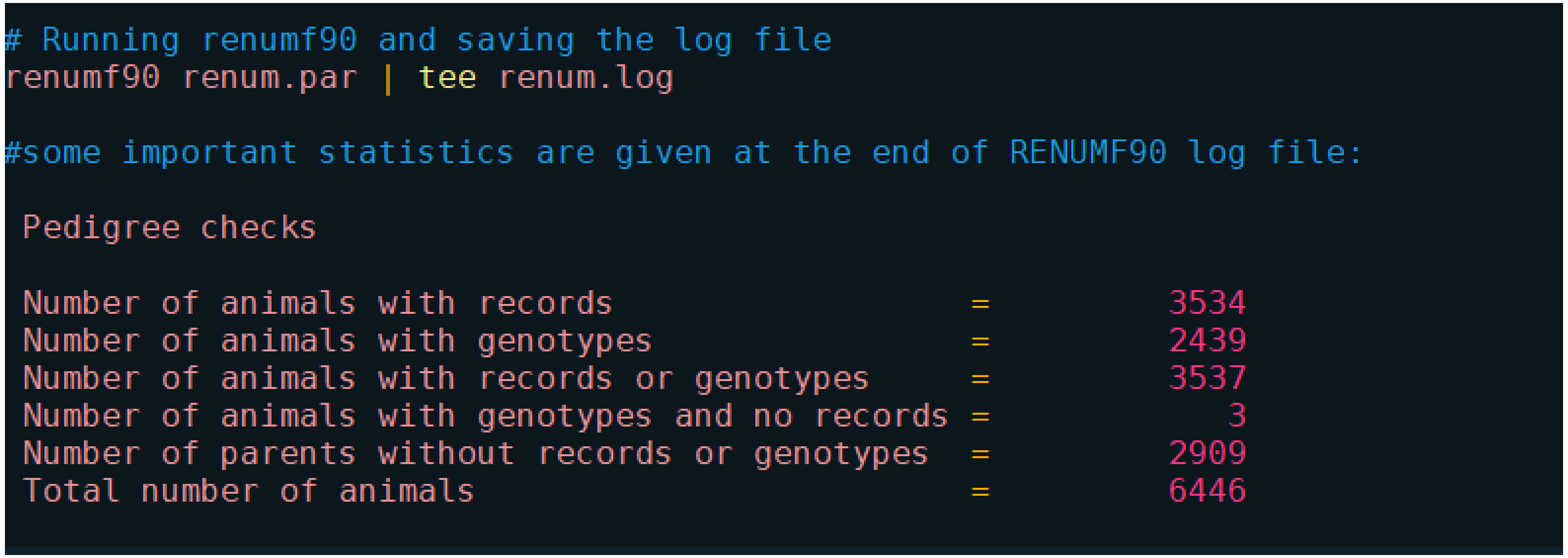
pedigree.txt :

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

genotypes.txt :

1: animal ID
2: marker information

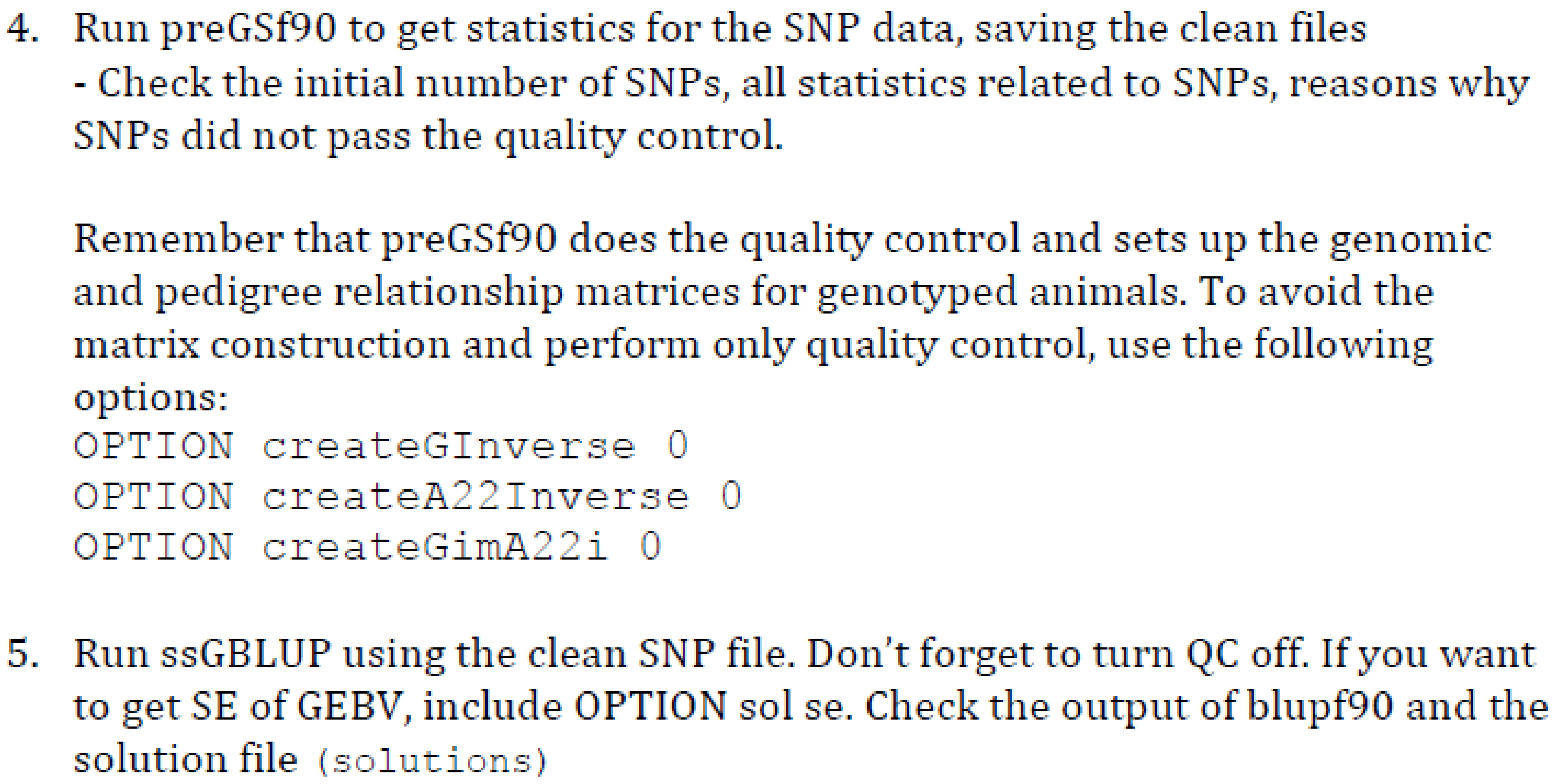
3. Run renumf90 in the single-trait context.



```
# Running renumf90 and saving the log file
renumf90 renum.par | tee renum.log

#some important statistics are given at the end of RENUMF90 log file:

 Pedigree checks

 Number of animals with records                =        3534
 Number of animals with genotypes              =        2439
 Number of animals with records or genotypes   =        3537
 Number of animals with genotypes and no records =         3
 Number of parents without records or genotypes  =      2909
 Total number of animals                       =        6446
```

This information in contained in the file we saved: renum.log

Block 2



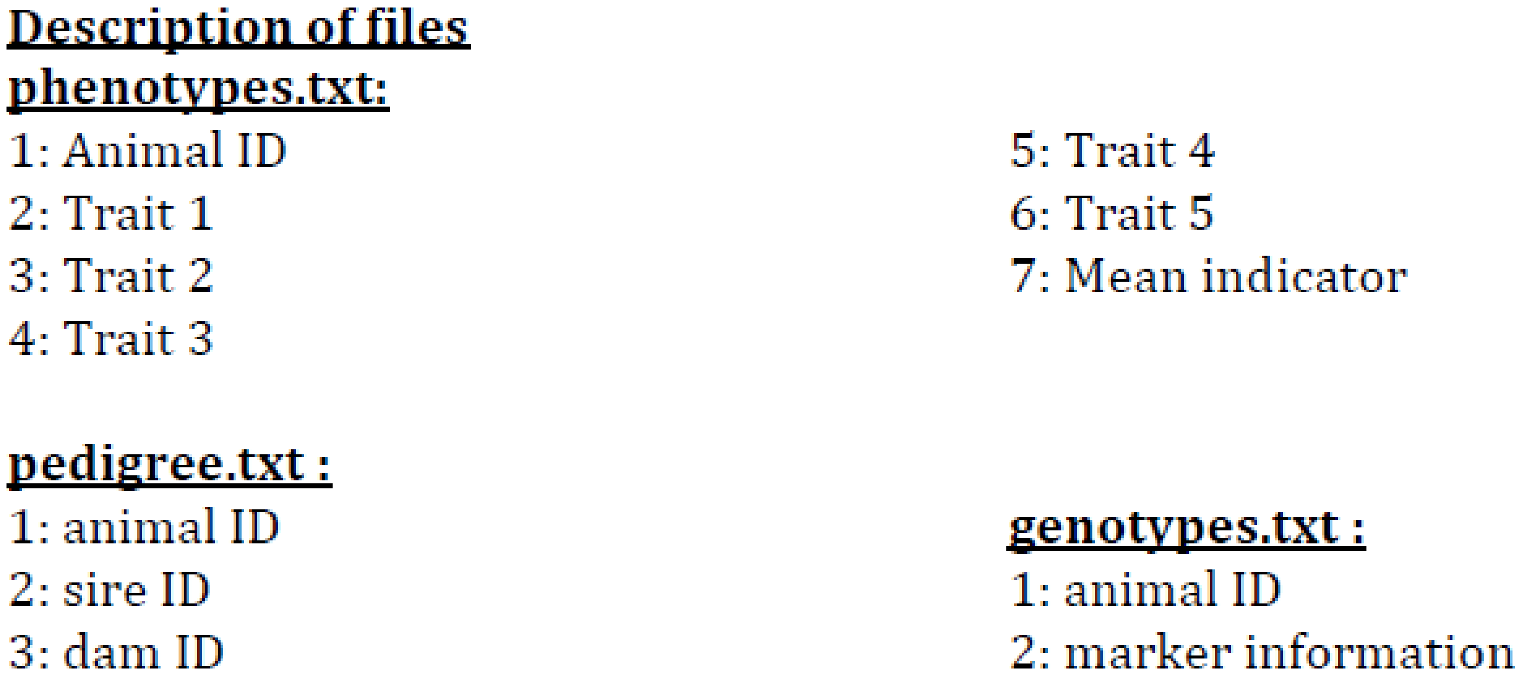
4. Run preGSf90 to get statistics for the SNP data, saving the clean files
   - Check the initial number of SNPs, all statistics related to SNPs, reasons why SNPs did not pass the quality control.

   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
   ```

5. 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)`

The dataset



Description of files

phenotypes.txt:

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

pedigree.txt :

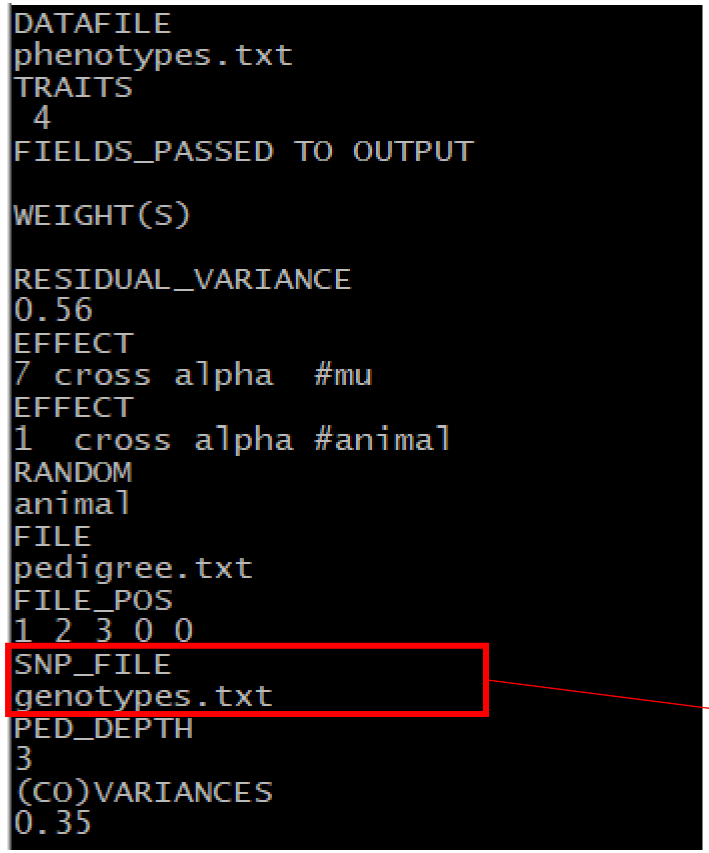
1: animal ID
2: sire ID
3: dam ID

genotypes.txt :

1: animal ID
2: marker information

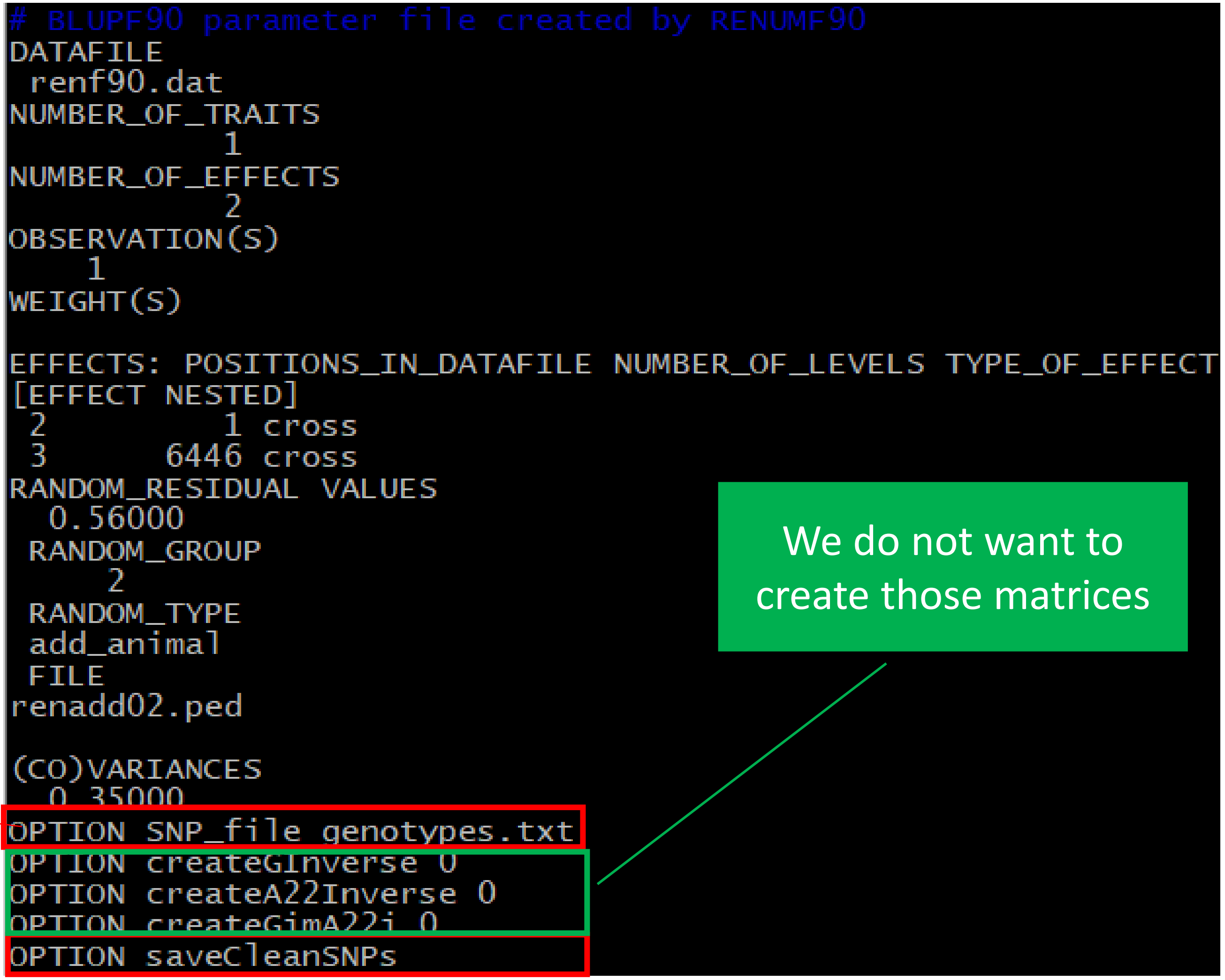
4. Run preGSf90 to get statistics for the SNP data, saving the clean files
   - Check the initial number of SNPs, all statistics related to SNPs, reasons why SNPs did not pass the quality control.

renum.par



```
DATAFILE
phenotypes.txt
TRAITS
 4
FIELDS_PASSED TO OUTPUT

WEIGHT(S)

RESIDUAL_VARIANCE
0.56
EFFECT
7 cross alpha  #mu
EFFECT
1  cross alpha #animal
RANDOM
animal
FILE
pedigree.txt
FILE_POS
1 2 3 0 0
SNP_FILE
genotypes.txt
PED_DEPTH
3
(CO)VARIANCES
0.35
```

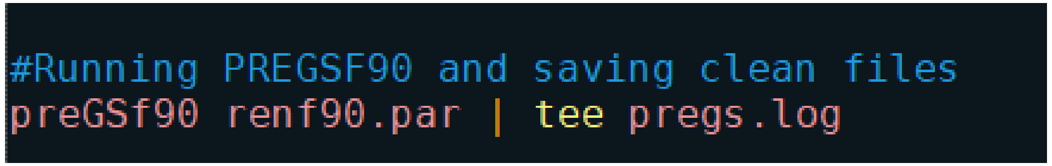
Renf90.par



```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT
[EFFECT NESTED]
 2         1 cross
 3      6446 cross
RANDOM_RESIDUAL VALUES
  0.56000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped

(CO)VARIANCES
  0.35000
OPTION SNP_file genotypes.txt
OPTION createGInverse 0
OPTION createA22Inverse 0
OPTION createGimA22i 0
OPTION saveCleanSNPs
```

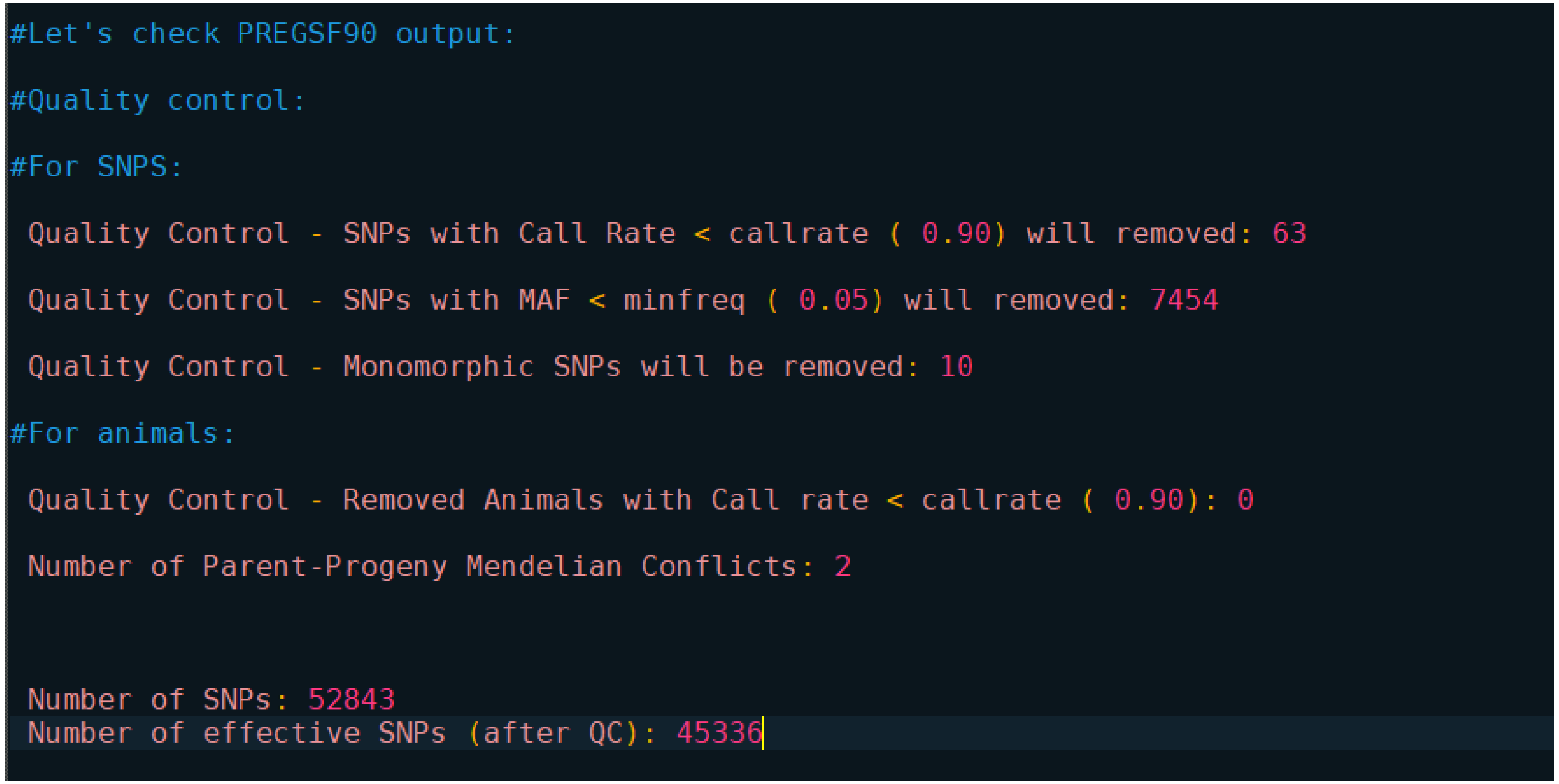
We do not want to create those matrices

4. 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.



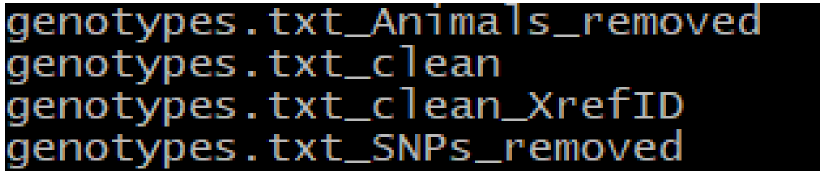
```
#Running PREGSF90 and saving clean files
preGSf90 renf90.par | tee pregs.log
```

pregs.log



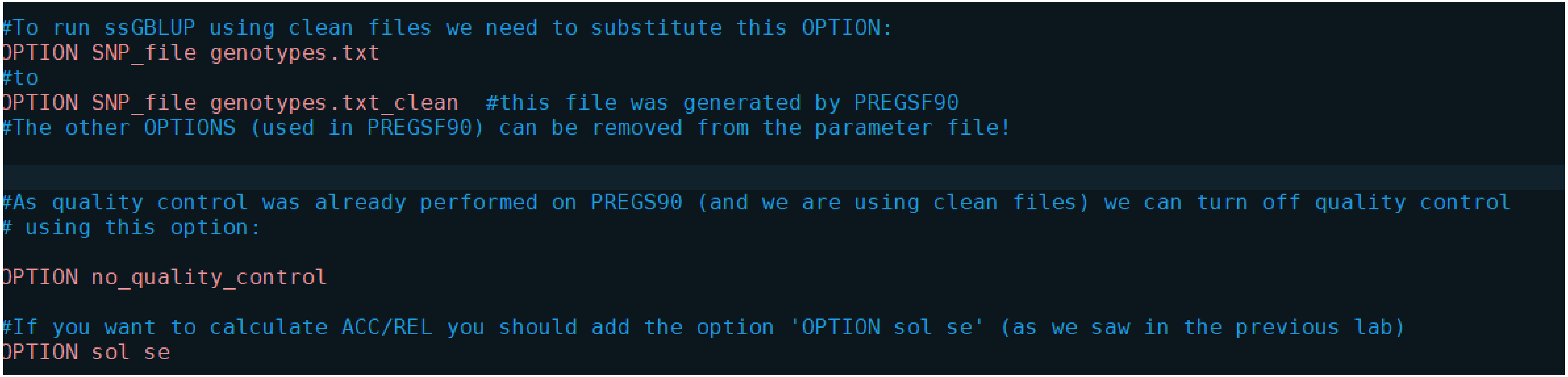
```
#Let's check PREGSF90 output:

#Quality control:

#For SNPS:

 Quality Control - SNPs with Call Rate < callrate ( 0.90) will removed: 63

 Quality Control - SNPs with MAF < minfreq ( 0.05) will removed: 7454

 Quality Control - Monomorphic SNPs will be removed: 10

#For animals:

 Quality Control - Removed Animals with Call rate < callrate ( 0.90): 0

 Number of Parent-Progeny Mendelian Conflicts: 2



 Number of SNPs: 52843
 Number of effective SNPs (after QC): 45336
```

Clean files and support files:



```
genotypes.txt_Animals_removed
genotypes.txt_clean
genotypes.txt_clean_XrefID
genotypes.txt_SNPs_removed
```

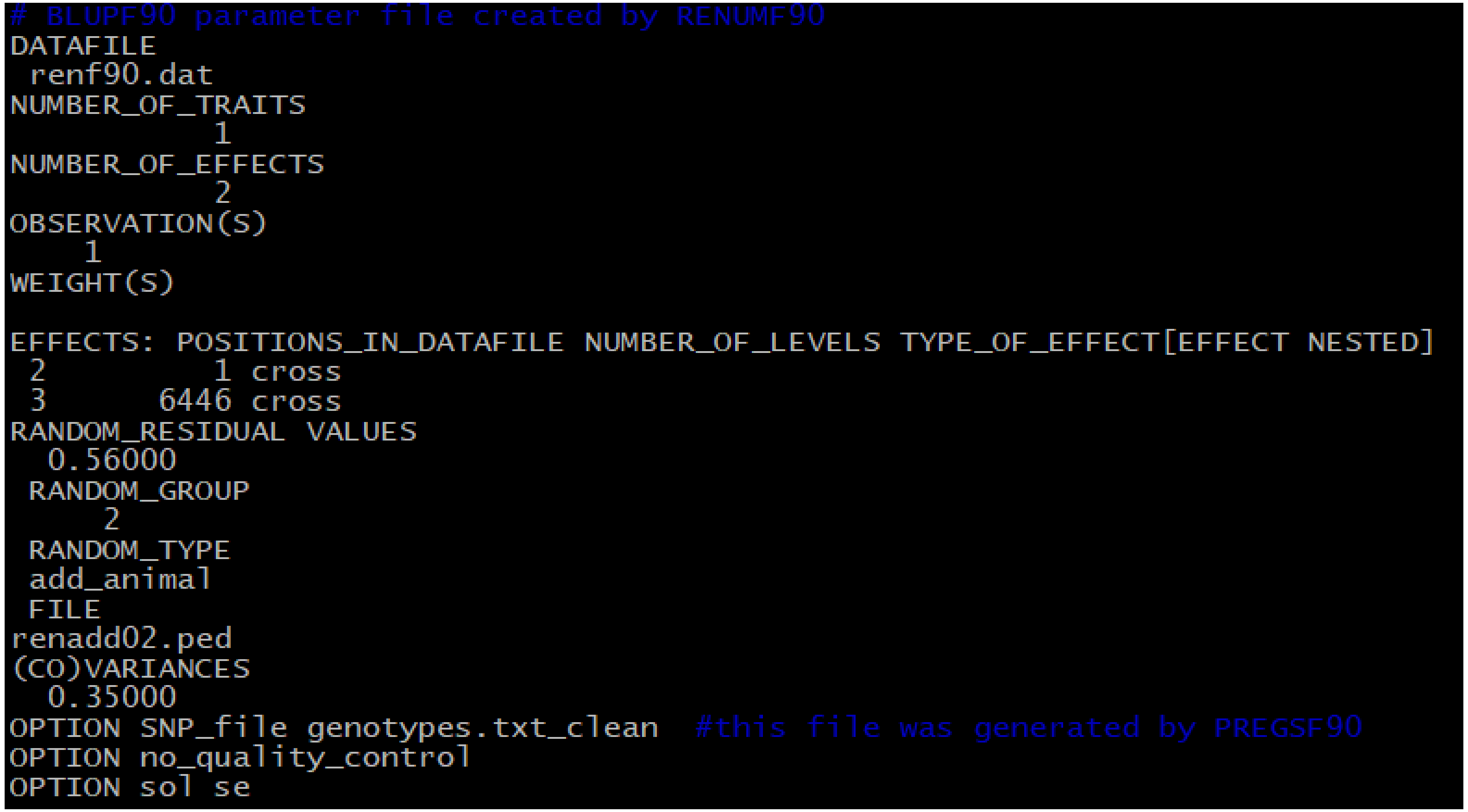
5. Run ssGBLUP using the clean SNP file. Don't forget to turn QC off. If you want to get SE of GEBV, include OPTION sol se. Check the output of blupf90 and the solution file `(solutions)`



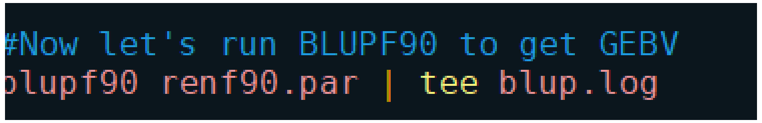
```
#To run ssGBLUP using clean files we need to substitute this OPTION:
OPTION SNP_file genotypes.txt
#to
OPTION SNP_file genotypes.txt_clean  #this file was generated by PREGSF90
#The other OPTIONS (used in PREGSF90) can be removed from the parameter file!


#As quality control was already performed on PREGS90 (and we are using clean files) we can turn off quality control
# using this option:

OPTION no_quality_control

#If you want to calculate ACC/REL you should add the option 'OPTION sol se' (as we saw in the previous lab)
OPTION sol se
```

5. 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)`

renf90.par

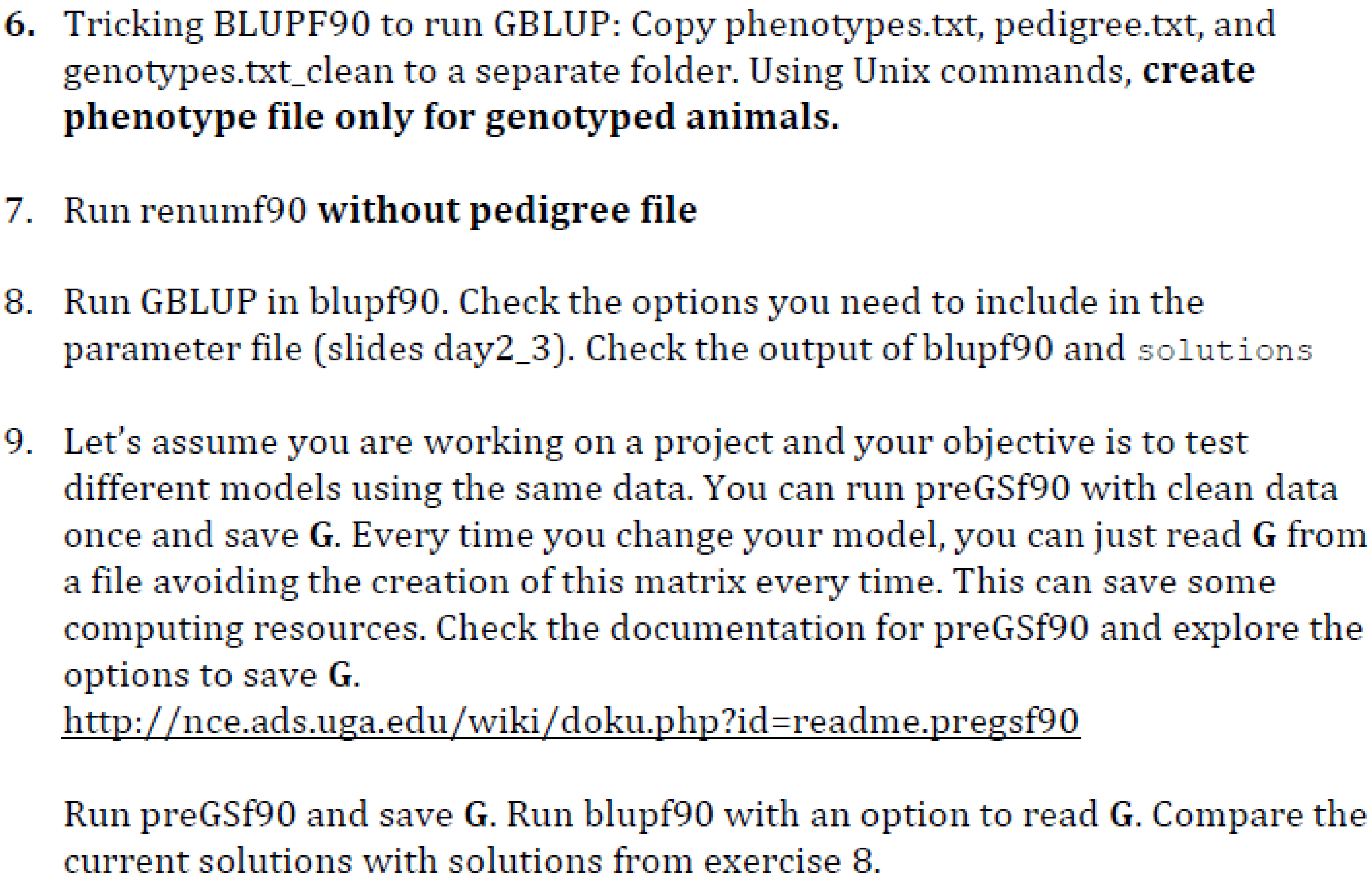


```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2          1 cross
 3       6446 cross
RANDOM_RESIDUAL VALUES
  0.56000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped
(CO)VARIANCES
  0.35000
OPTION SNP_file genotypes.txt_clean  #this file was generated by PREGSF90
OPTION no_quality_control
OPTION sol se
```



```
#Now let's run BLUPF90 to get GEBV
blupf90 renf90.par | tee blup.log
```

Block 3



6. Tricking BLUPF90 to run GBLUP: Copy phenotypes.txt, pedigree.txt, and genotypes.txt_clean to a separate folder. Using Unix commands, **create phenotype file only for genotyped animals.**

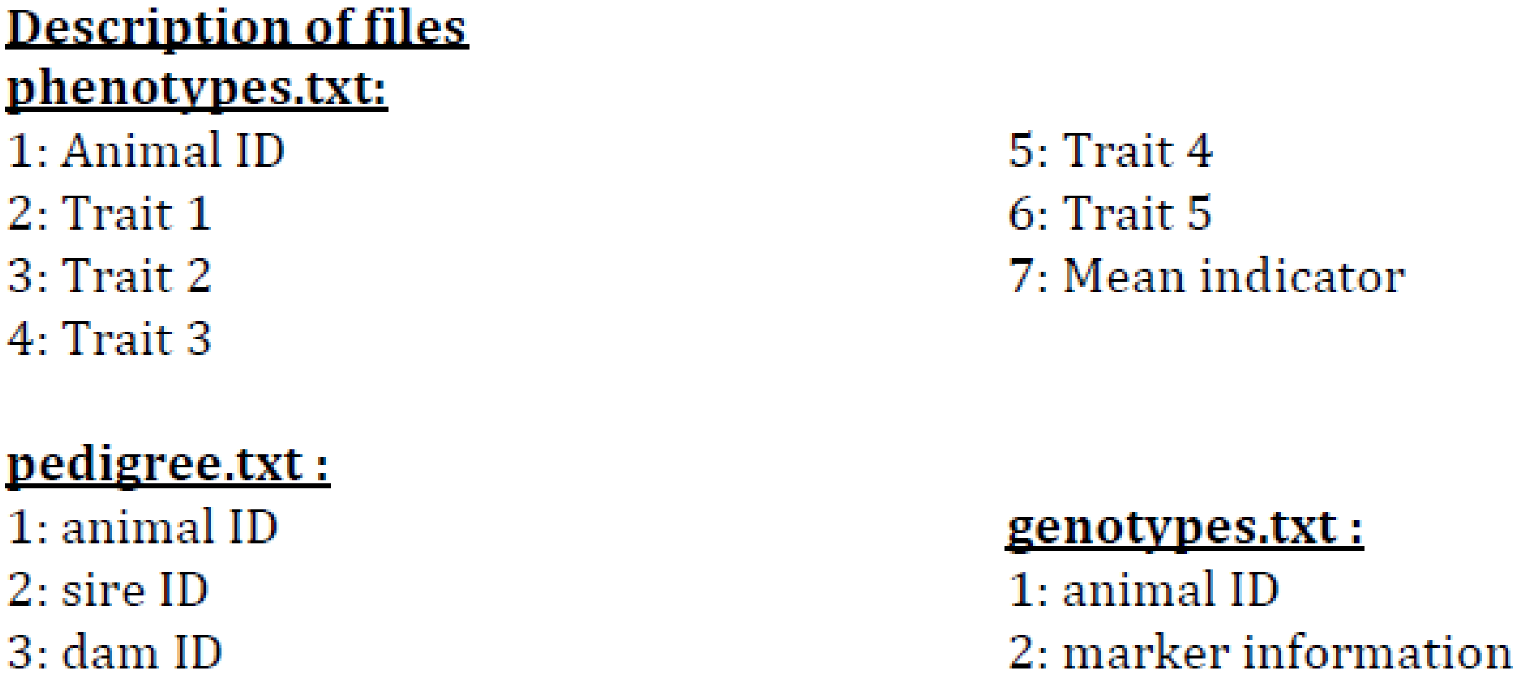
7. Run renumf90 **without pedigree file**

8. 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`

9. 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.

The data set



Description of files

phenotypes.txt:

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

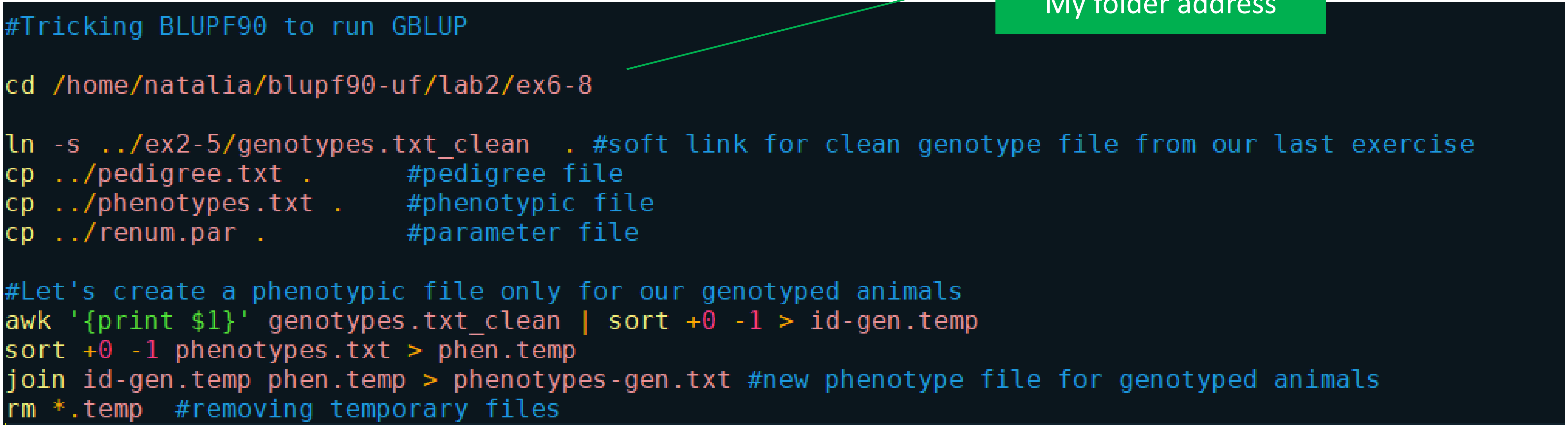
pedigree.txt :

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

genotypes.txt :

1: animal ID
2: marker information

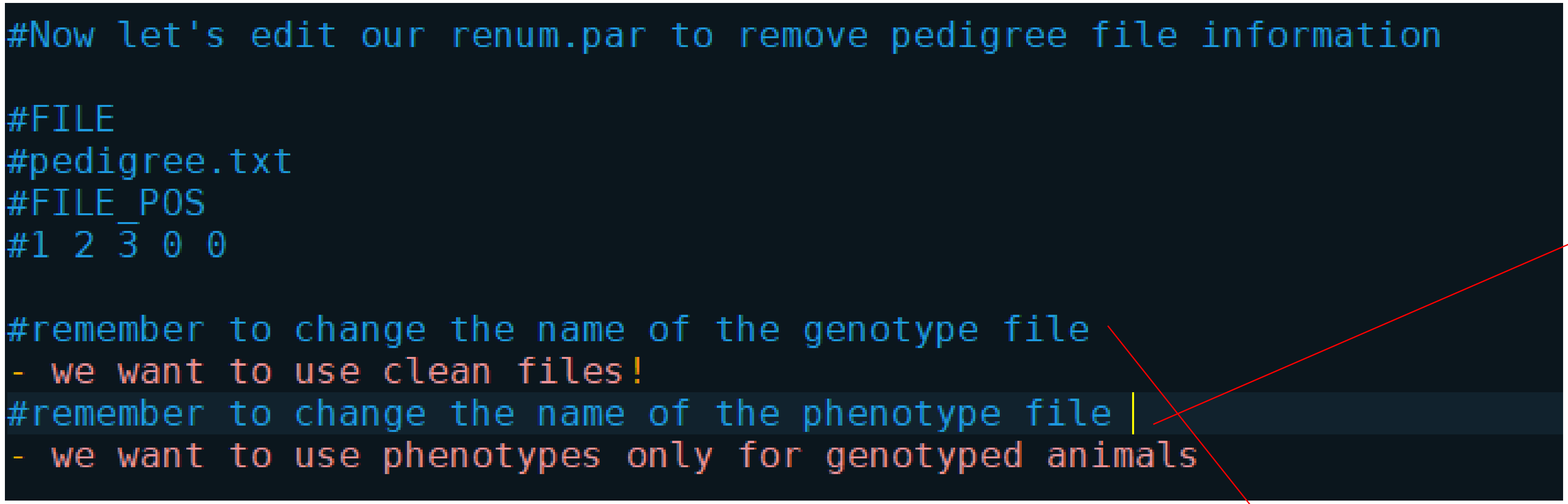
6. Tricking BLUPF90 to run GBLUP: Copy phenotypes.txt, pedigree.txt, and genotypes.txt_clean to a separate folder. Using Unix commands, **create phenotype file only for genotyped animals.**



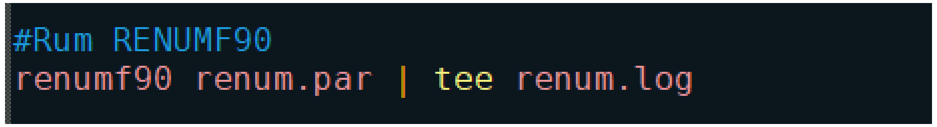
```
#Tricking BLUPF90 to run GBLUP

cd /home/natalia/blupf90-uf/lab2/ex6-8

ln -s ../ex2-5/genotypes.txt_clean  . #soft link for clean genotype file from our last exercise
cp ../pedigree.txt .       #pedigree file
cp ../phenotypes.txt .     #phenotypic file
cp ../renum.par .          #parameter file

#Let's create a phenotypic file only for our genotyped animals
awk '{print $1}' genotypes.txt_clean | sort +0 -1 > id-gen.temp
sort +0 -1 phenotypes.txt > phen.temp
join id-gen.temp phen.temp > phenotypes-gen.txt #new phenotype file for genotyped animals
rm *.temp  #removing temporary files
```

My folder address

7. Run renumf90 **without pedigree file**

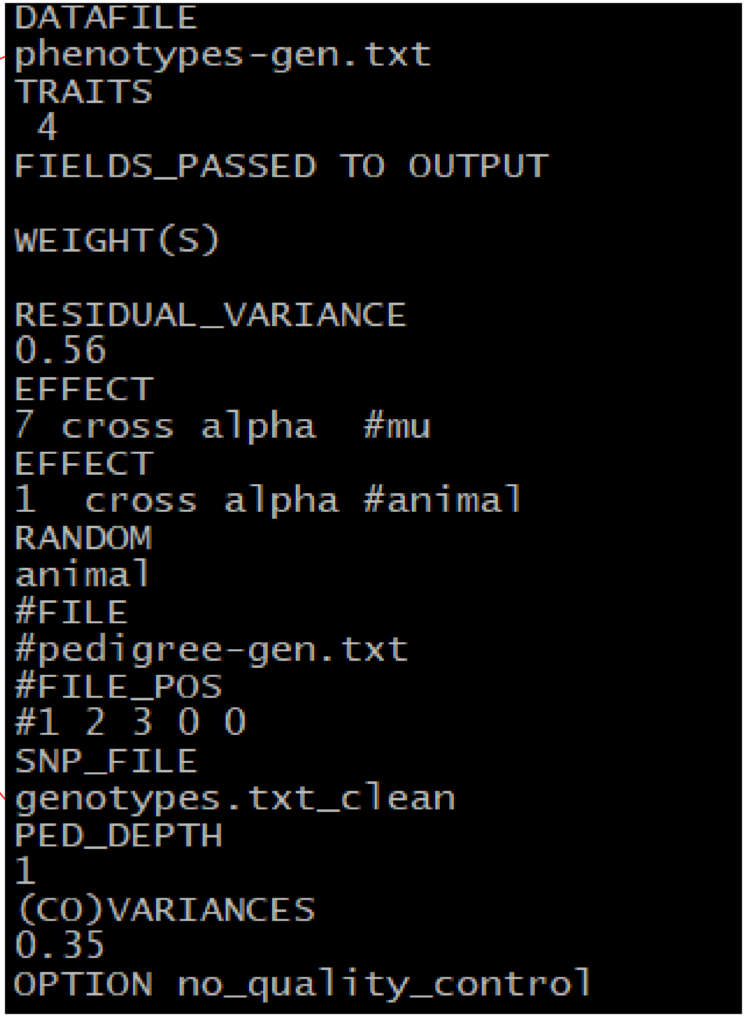


```
#Now let's edit our renum.par to remove pedigree file information

#FILE
#pedigree.txt
#FILE_POS
#1 2 3 0 0

#remember to change the name of the genotype file
- we want to use clean files!
#remember to change the name of the phenotype file
- we want to use phenotypes only for genotyped animals
```



```
#Rum RENUMF90
renumf90 renum.par | tee renum.log
```

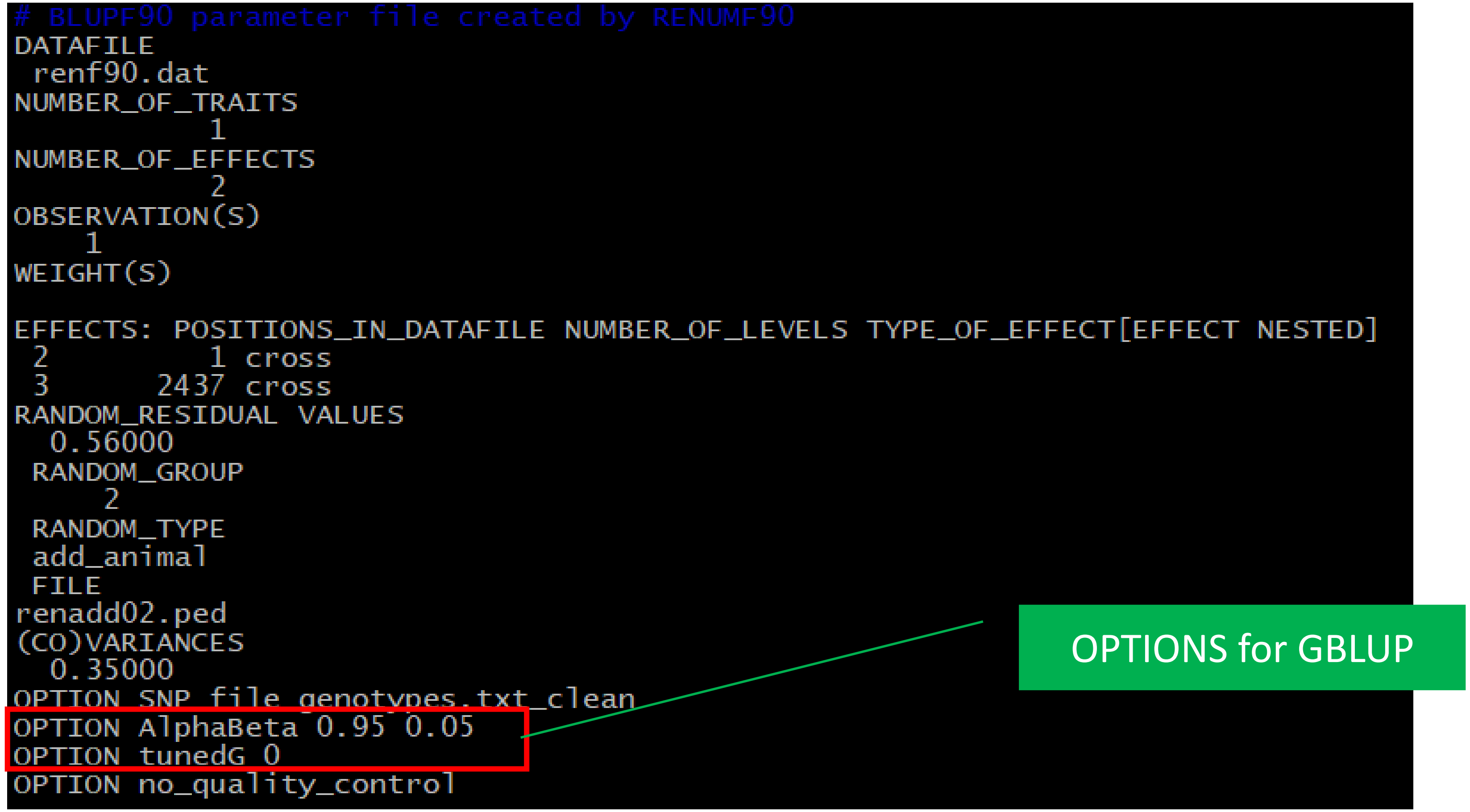
renum.par



```
DATAFILE
phenotypes-gen.txt
TRAITS
 4
FIELDS_PASSED TO OUTPUT

WEIGHT(S)

RESIDUAL_VARIANCE
0.56
EFFECT
7 cross alpha  #mu
EFFECT
1  cross alpha #animal
RANDOM
animal
#FILE
#pedigree-gen.txt
#FILE_POS
#1 2 3 0 0
SNP_FILE
genotypes.txt_clean
PED_DEPTH
1
(CO)VARIANCES
0.35
OPTION no_quality_control
```

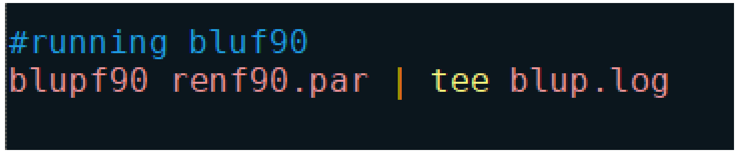
8. Run GBLUP in blupf90. Check the options you need to include in the parameter file (slides day2_3). Check the output of blupf90 and `solutions`

Renf90.par

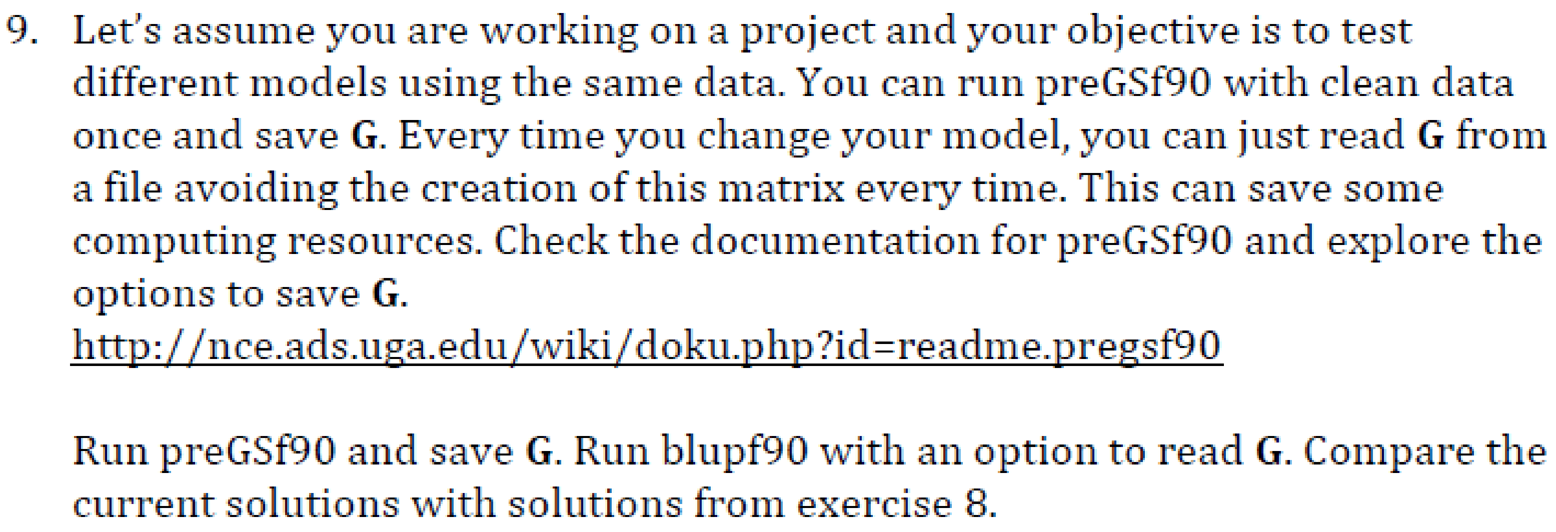


```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         1 cross
 3      2437 cross
RANDOM_RESIDUAL VALUES
  0.56000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped
(CO)VARIANCES
  0.35000
OPTION SNP_file genotypes.txt_clean
OPTION AlphaBeta 0.95 0.05
OPTION tunedG 0
OPTION no_quality_control
```

OPTIONS for GBLUP

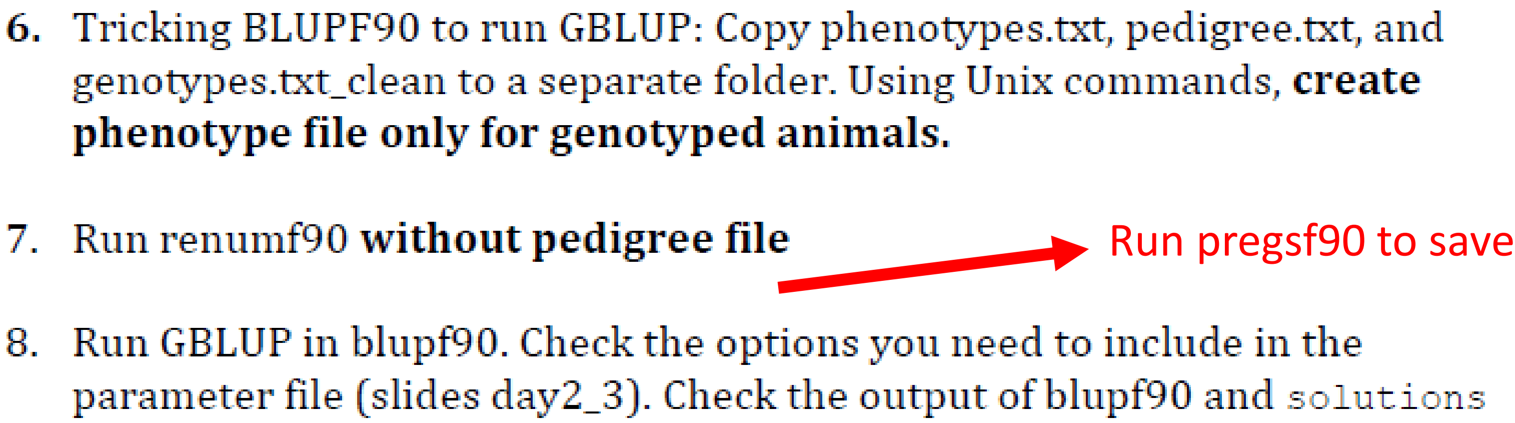


```
#running bluf90
blupf90 renf90.par | tee blup.log
```



9. 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.



6. Tricking BLUPF90 to run GBLUP: Copy phenotypes.txt, pedigree.txt, and genotypes.txt_clean to a separate folder. Using Unix commands, **create phenotype file only for genotyped animals.**

7. Run renumf90 **without pedigree file**

Run pregsf90 to save the matrix G

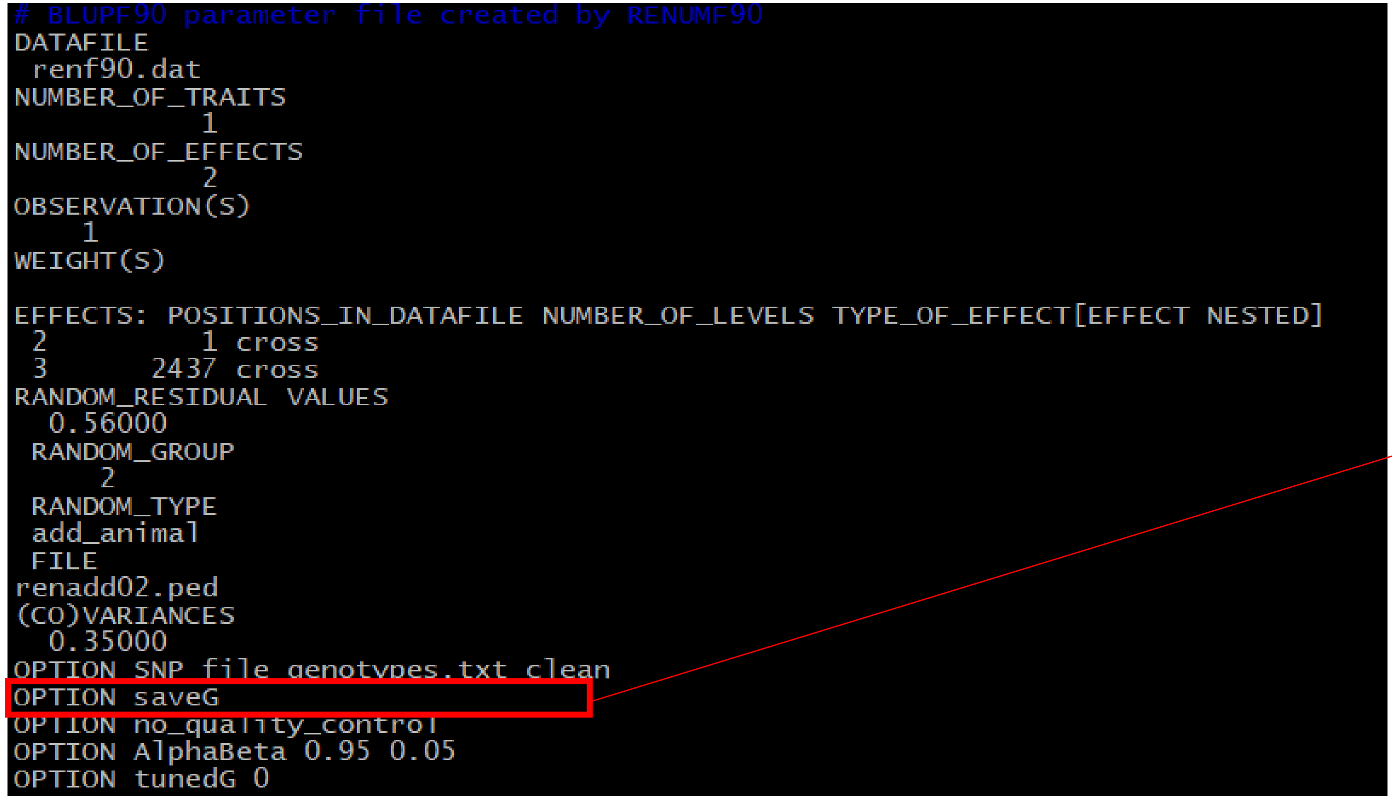
8. Run GBLUP in blupf90. Check the options you need to include in the parameter file (slides day2_3). Check the output of blupf90 and `solutions`

Run blupf90 with the saved G

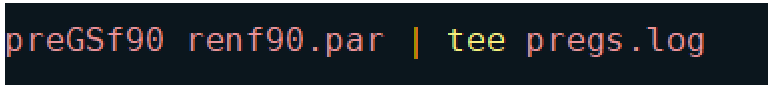
9. Let's assume you are working on a project and your objective is to test different models using the same data. You can run preGSf90 with clean data once and save **G**. Every time you change your model, you can just read **G** from a file avoiding the creation of this matrix every time. This can save 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.

Renf90.par

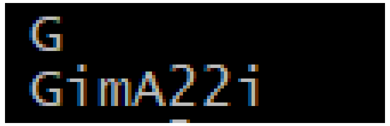


```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         1 cross
 3      2437 cross
RANDOM_RESIDUAL VALUES
  0.56000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped
(CO)VARIANCES
  0.35000
OPTION SNP_file genotypes.txt_clean
OPTION saveG
OPTION no_quality_control
OPTION AlphaBeta 0.95 0.05
OPTION tunedG 0
```



```
preGSf90 renf90.par | tee pregs.log
```

Files saved by pregsf90:

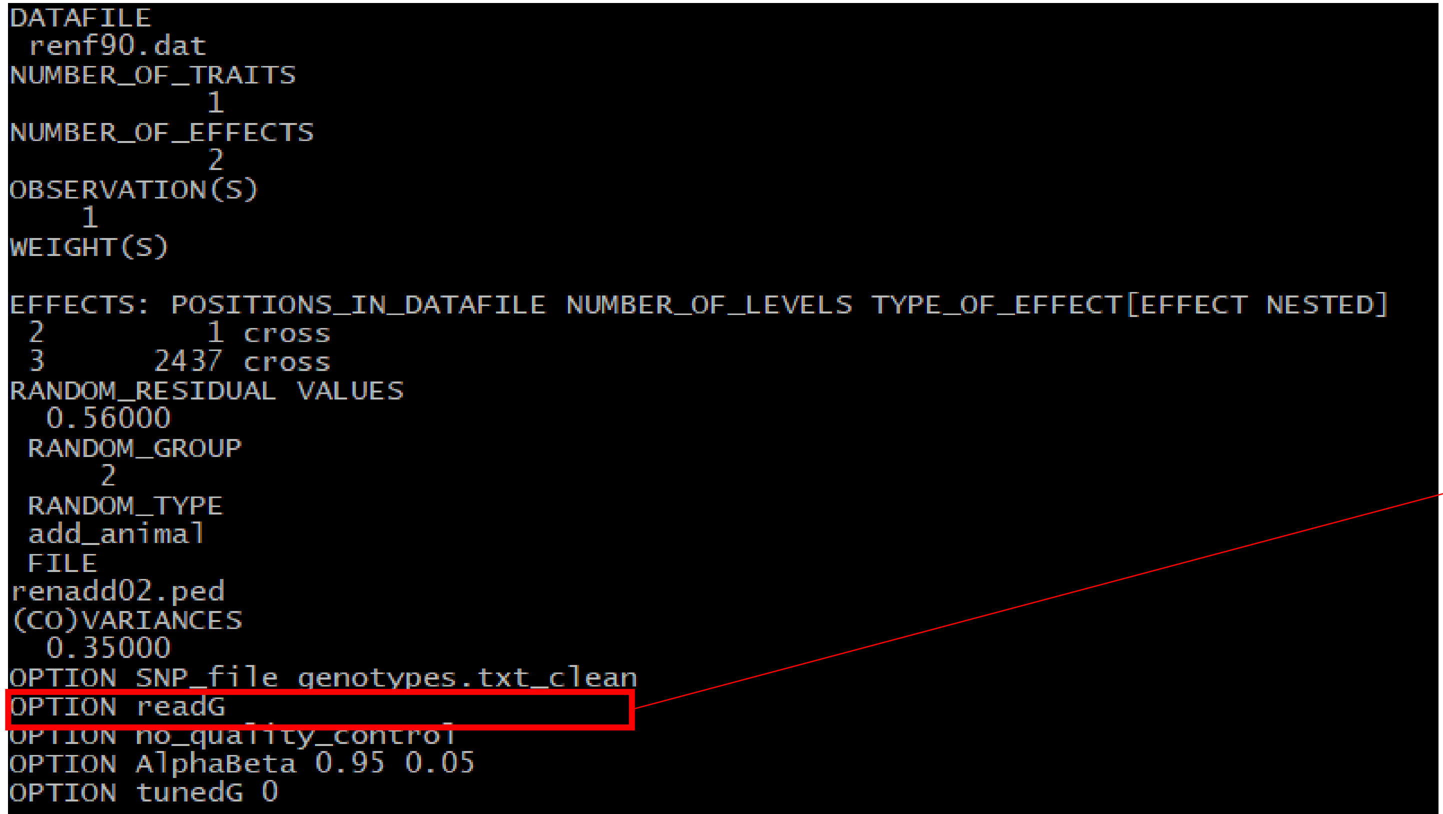


```
G
GimA22i
```

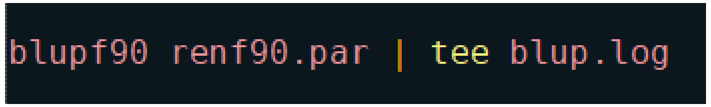
9. Let's assume you are working on a project and your objective is to test different models using the same data. You can run preGSf90 with clean data once and save **G**. Every time you change your model, you can just read **G** from a file avoiding the creation of this matrix every time. This can save 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.

Renf90.par

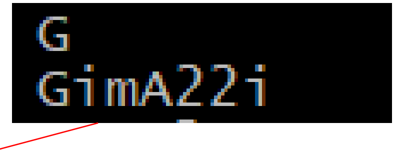


```
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         1 cross
 3      2437 cross
RANDOM_RESIDUAL VALUES
  0.56000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped
(CO)VARIANCES
  0.35000
OPTION SNP_file genotypes.txt_clean
OPTION readG
OPTION no_quality_control
OPTION AlphaBeta 0.95 0.05
OPTION tunedG 0
```



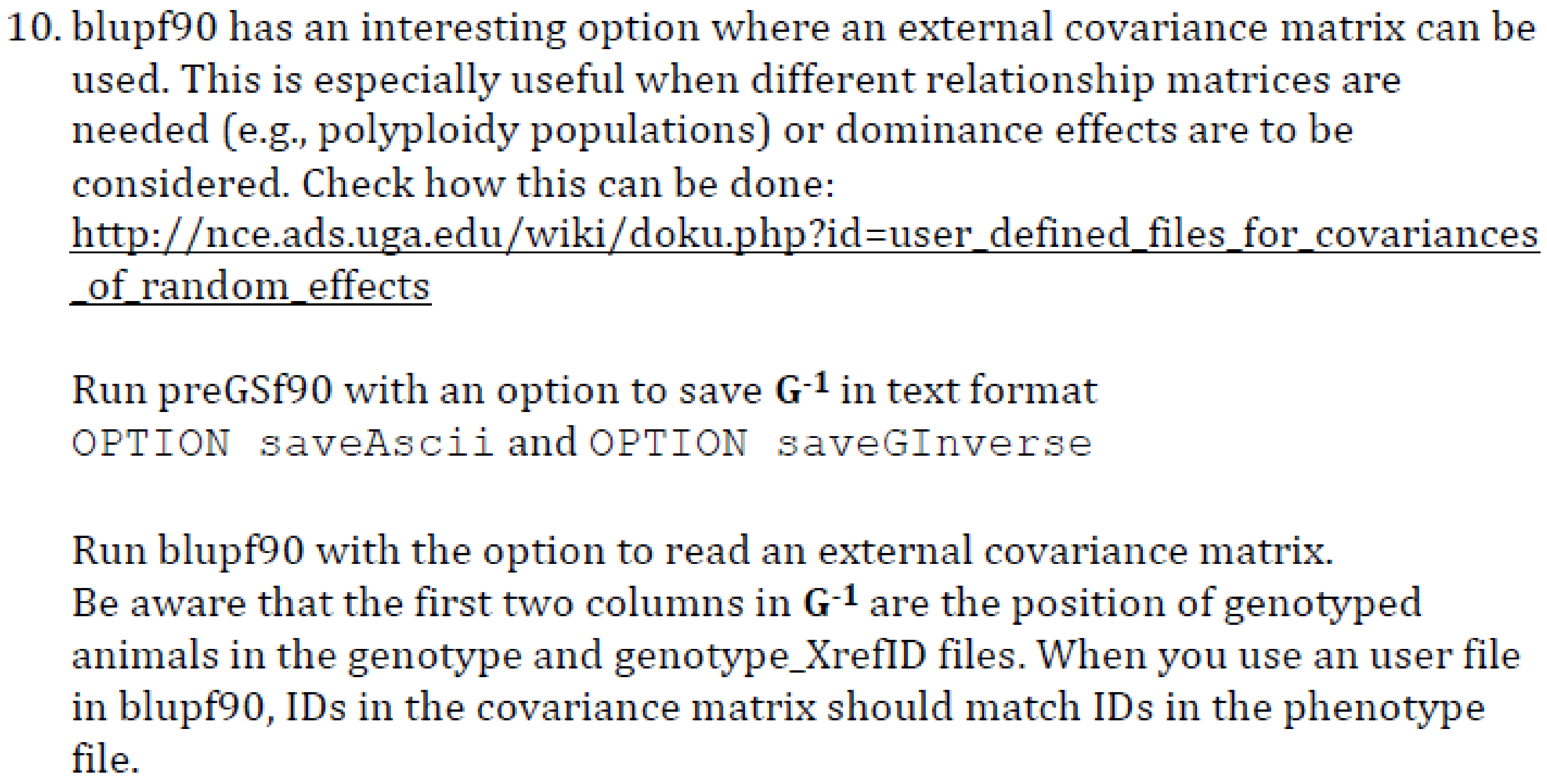
```
blupf90 renf90.par | tee blup.log
```

Files saved by pregsf90:



```
G
GimA22i
```

Block 4



10. 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

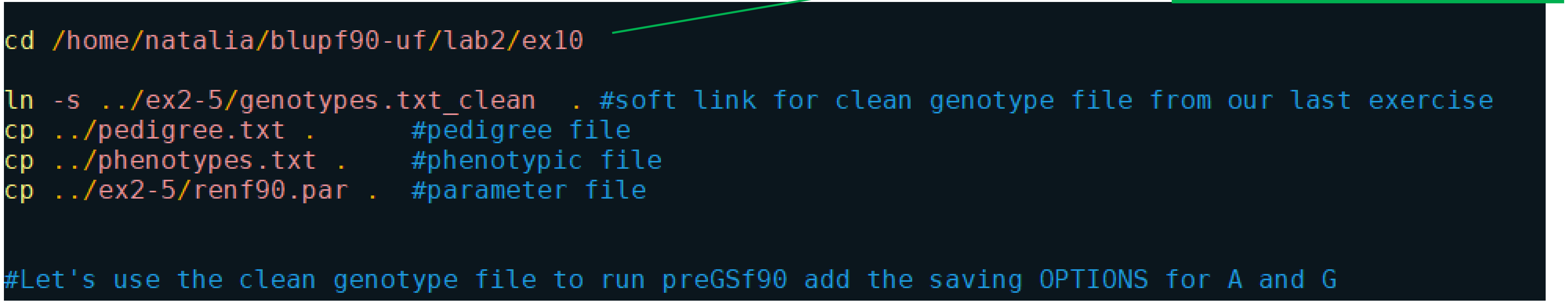
    Run preGSf90 with an option to save $\mathbf{G}^{-1}$ in text format
    `OPTION saveAscii` and `OPTION saveGInverse`

    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.

10. 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`

My folder address

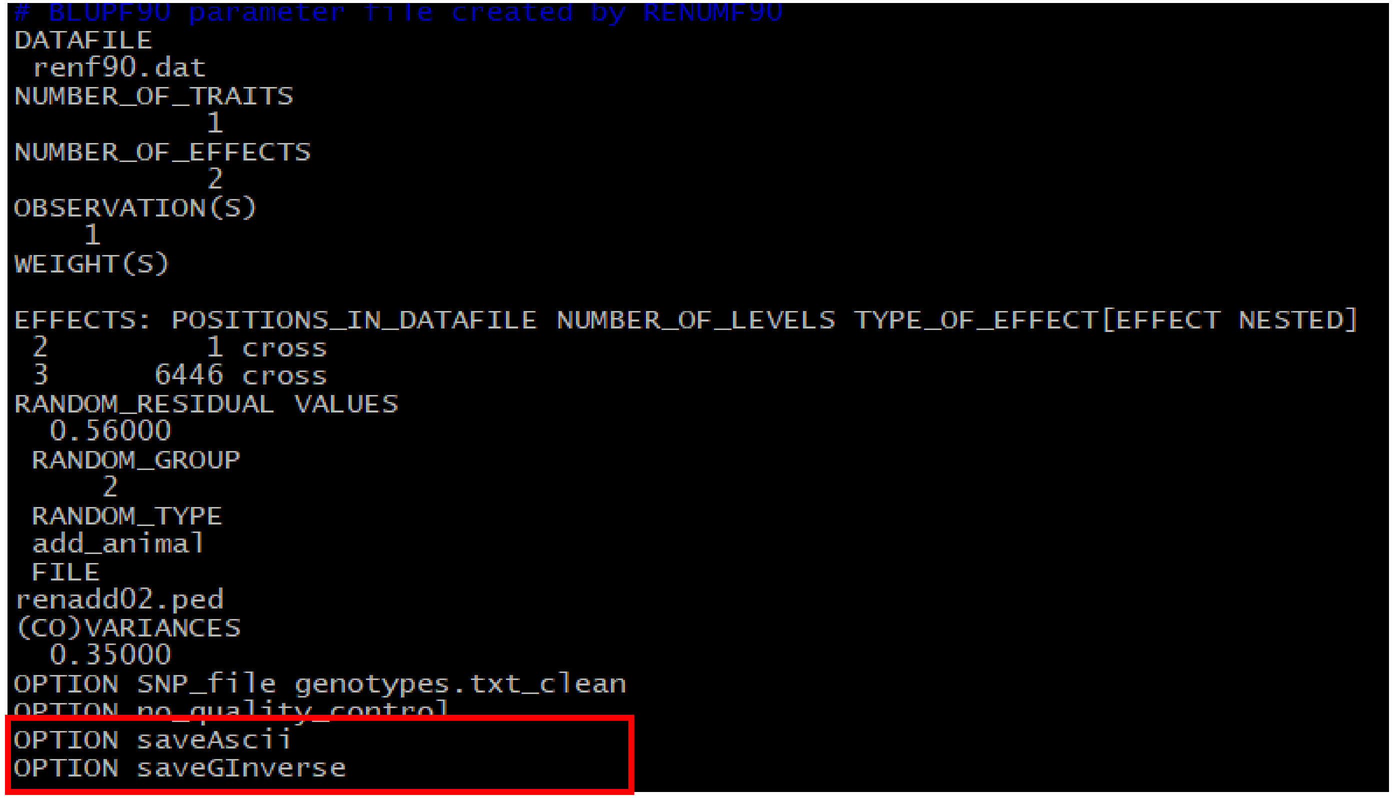


```
cd /home/natalia/blupf90-uf/lab2/ex10

ln -s ../ex2-5/genotypes.txt_clean  . #soft link for clean genotype file from our last exercise
cp ../pedigree.txt .      #pedigree file
cp ../phenotypes.txt .    #phenotypic file
cp ../ex2-5/renf90.par .  #parameter file


#Let's use the clean genotype file to run preGSf90 add the saving OPTIONS for A and G
```

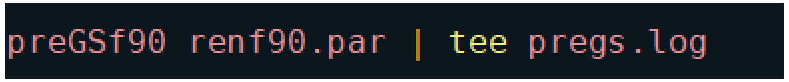
10. blupf90 has an interesting option where an external covariance matrix can be used. This is especially useful when different relationship matrices are needed (e.g., polyploidy populations) or dominance effects are to be considered. Check how this can be done: http://nce.ads.uga.edu/wiki/doku.php?id=user_defined_files_for_covariances_of_random_effects

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

Renf90.par

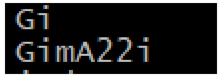


```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         1 cross
 3      6446 cross
RANDOM_RESIDUAL VALUES
  0.56000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped
(CO)VARIANCES
  0.35000
OPTION SNP_file genotypes.txt_clean
OPTION no_quality_control
OPTION saveAscii
OPTION saveGInverse
```

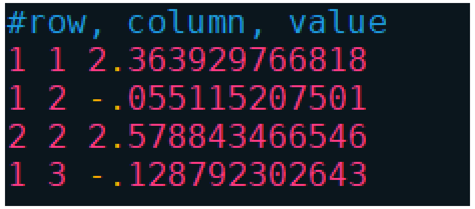


```
preGSf90 renf90.par | tee pregs.log
```

Files saved by pregsf90:

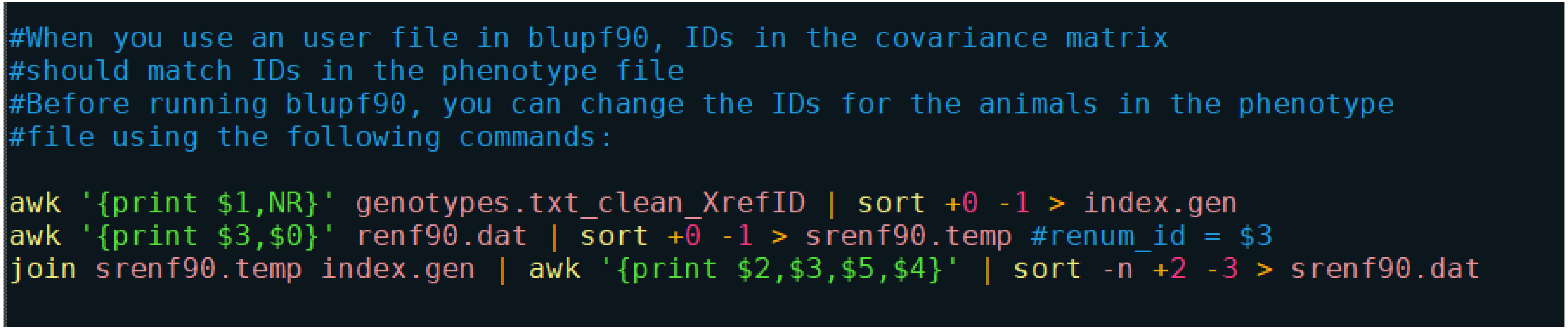


```
Gi
GimA22i
```



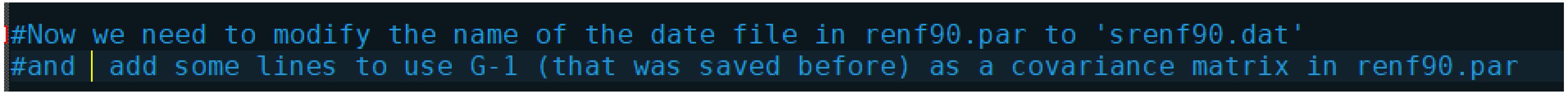
```
#row, column, value
1 1 2.363929766818
1 2 -.055115207501
2 2 2.578843466546
1 3 -.128792302643
```

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.

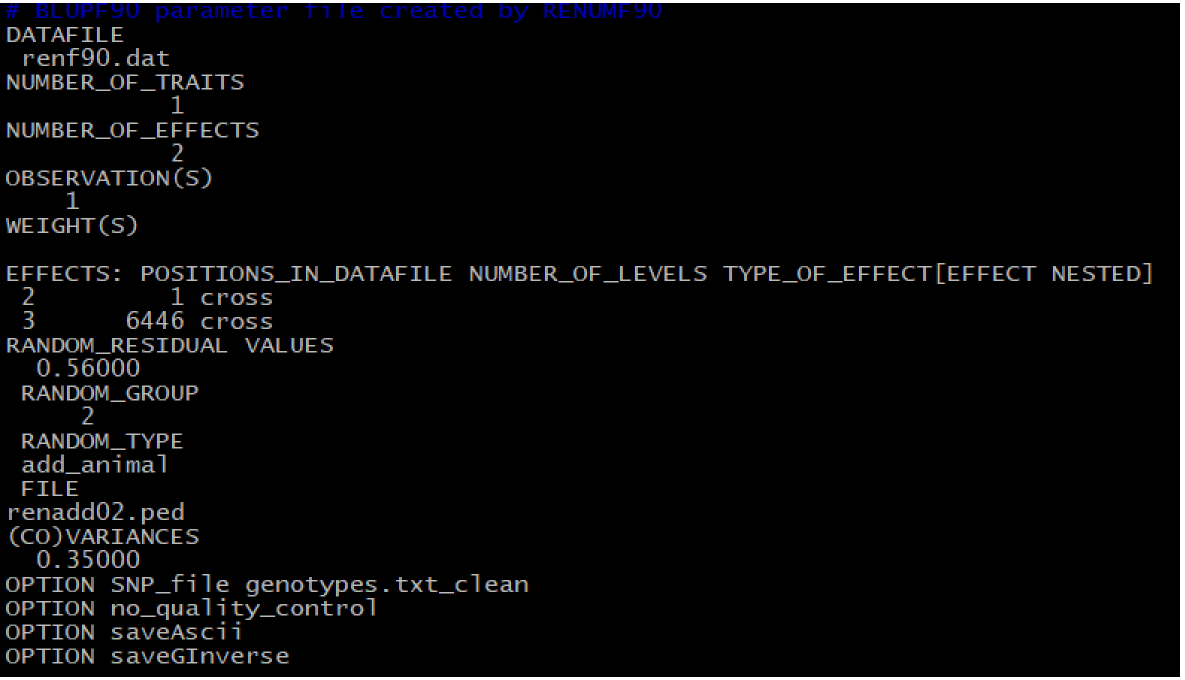


```
#When you use an user file in blupf90, IDs in the covariance matrix
#should match IDs in the phenotype file
#Before running blupf90, you can change the IDs for the animals in the phenotype
#file using the following commands:

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

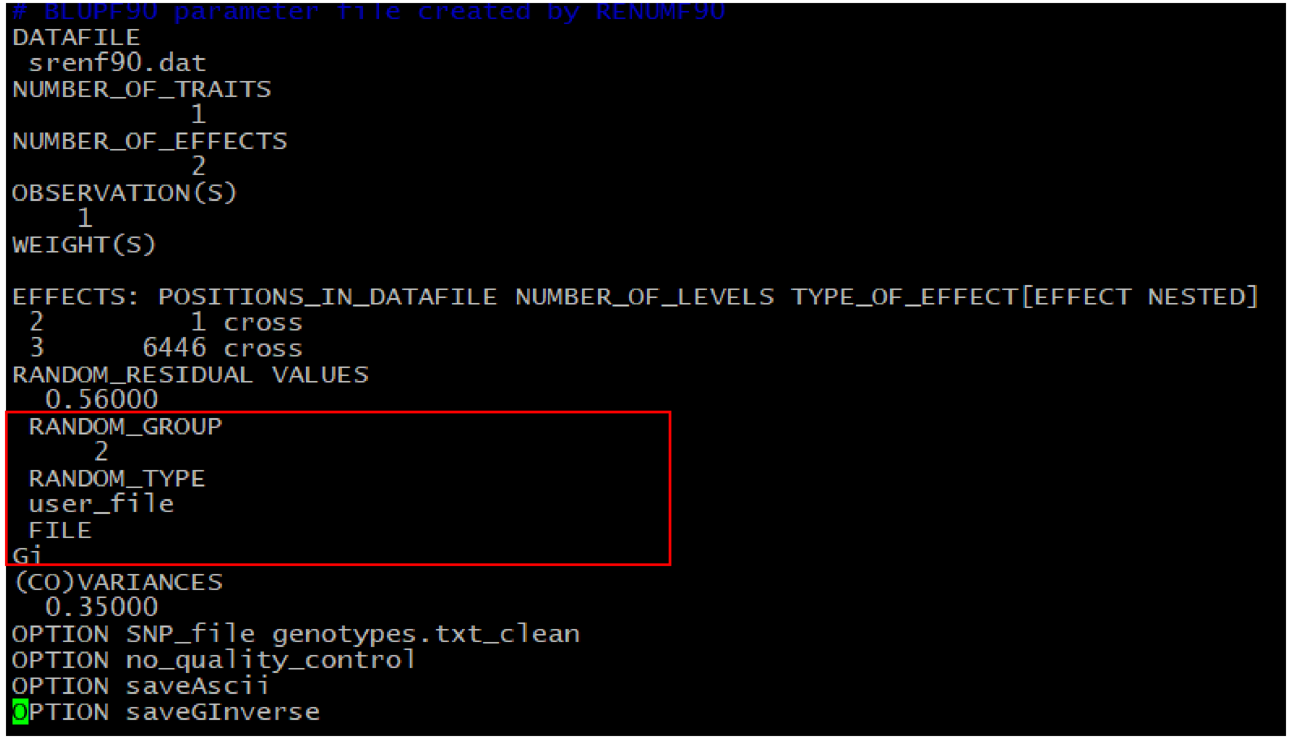
Run blupf90 with the option to read an external covariance matrix. Be aware that the first two columns in **G**$^{-1}$ are the position of genotyped animals in the genotype and genotype_XrefID files. When you use an user file in blupf90, IDs in the covariance matrix should match IDs in the phenotype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



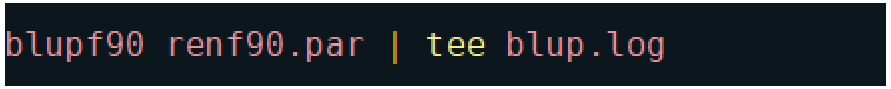
```
#Now we need to modify the name of the date file in renf90.par to 'srenf90.dat'
#and add some lines to use G-1 (that was saved before) as a covariance matrix in renf90.par
```



```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         1 cross
 3      6446 cross
RANDOM_RESIDUAL VALUES
  0.56000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_animal
 FILE
renadd02.ped
(CO)VARIANCES
  0.35000
OPTION SNP_file genotypes.txt_clean
OPTION no_quality_control
OPTION saveAscii
OPTION saveGInverse
```



```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 srenf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         1 cross
 3      6446 cross
RANDOM_RESIDUAL VALUES
  0.56000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 user_file
 FILE
Gi
(CO)VARIANCES
  0.35000
OPTION SNP_file genotypes.txt_clean
OPTION no_quality_control
OPTION saveAscii
OPTION saveGInverse
```



```
blupf90 renf90.par | tee blup.log
```