Genome wide complex trait analysis (GCTA)

GCTA software is a command line tool that has the most support on the Linux operating system. The GCTA program uses an argument interface similar to PLINK. It is advised that you visit GCTA's website http://cnsgenomics.com/software/gcta/ and familiarise yourself a little with the software (perhaps for five minutes). GCTA has many functions but one of its primary uses is variance component estimation via restricted maximum likelihood (REML).

GCTA is available for Linux, MAC and Windows environments. We assume that you put the data in a 'Data/PartE/' directory and put the gcta software in your current directory. You should create a new 'Results/' folder where you will store the output of this practical. For Windows users, you can also add the 'bin/WINDOWS' folder to your PATH.

Similarly to previous practicals, you will most likely use GCTA in a Linux environment (e.g. HPC), thus all command lines in this material are for Linux users. Commands for this practical can be found in the 'linux_MAC_commands.txt' or 'windows_commands.txt' files.

Data

We will use the genotype and phenotype data, which includes height and serum transferrin levels. These data will be in `PLINK` binary format. We will first attempt to use the SNP marker data to build a genetic relationship matrix and estimate the proportion of phenotypic variance explained by genome wide SNPs.

1 GCTA introduction

In this practical we will use the GCTA software [Yang et al., 2011, 2010] to estimate variance components, where our data will be SNP markers from the whole genome or a chromosome. We will take a look at some of GCTA's primary functions – building the SNP genetic relationship matrix (GRM), and variance component estimation. In this practical we will use `GCTA` to do genomic restricted maximum likelihood to estimate variance components (GREML). This will be done for both the height and serum transferrin levels data and we will then interrogate and compare these Results using R.

We begin with building the GRM. If you are interested in the conceptual basis for the GRM built with GCTA please refer to [Yang et al., 2010]. Peak RAM usage for building the GRM is high and thus it is likely that your process may fail if your computer does not have enough resources (around 8GB of RAM). If your system does not have this amount of RAM, we will use a much smaller subset of the data to build the GRM as an exercise; the real GRM is already stored in your `data` folder and will be used in subsequent analyses. If you need help with the syntax for `GCTA` please take a look at the url `<http://cnsgenomics.com/software/gcta/>`.

We will not attempt to build the GRM of Listing 1 and 2 as the run time on a good computer is approximately ten minutes. If you would like to build the GRM in your spare time, Listings 1 and 2 should work on an 8GB RAM machine and a 2GB RAM machine respectively.

```
./gcta64 --bfile Data/PartE/optional/QIMRX_cleaned --make-grm --autosome --out
    Results/QIMRX --thread-num 2

./gcta64 --grm Results/QIMRX --grm-cutoff 0.8 --make-grm --out Data/PartE/QIMRX_no_
    twin
```

Listing 1: *Building a GRM with `GCTA` - do not run during the practical*

```
./gcta64  --bfile Data/PartE/optional/QIMRX_cleaned_small --make-grm --autosome \
          --out Results/QIMRX_small
```

Listing 2: *Building a smaller GRM with `GCTA` - do not run during the practical*

`GCTA` prints output to the console whilst processing the GRM and saves this information to a `.log` file. Some of the key summary statistics of the GRM built above are seen below.

```
$ Summary of the GRM:
$ Mean of diagonals = 1.00083
$ Variance of diagonals = 9.96466e-05
$ Mean of off-diagonals = -0.000206112
$ Variance of off-diagonals = 4.47574e-05
```

Listing 3: *Summary of GRM*

This process generates a binary file `QIMRX.grm.bin` with auxiliary file `QIMRX.grm.N.bin` (or two files `QIMRX.grm.gz` and `QIMRX.grm.id` for older versions on Mac and Windows). Let's take a look at these files in R using a function borrowed from the 'OmicKriging' R package.

```
> # Read in the gzipped GRM file
> # will need to use readGRM functions
> grm <- read_GRMBin("Data/PartE/QIMRX_no_twin.grm")
> grm[1:5,1:5]
```

```
             883           884          885           886          887
883  0.982474327  0.4307622612  0.001788882 -0.0012868681 -0.002608952
884  0.430762261  0.9971557856  0.001014439 -0.0002567511 -0.001598181
885  0.001788882  0.0010144389  1.000037909 -0.0037154893  0.003178963
886 -0.001286868 -0.0002567511 -0.003715489  1.0015679598  0.002775257
887 -0.002608952 -0.0015981813  0.003178963  0.0027752570  1.007973433
```

2 Estimating the proportion of phenotypic variation due to genome-wide SNPs using GCTA

Exercise 1.
- What is the percentage of phenotypic variance that is explained by common SNPs for height and serum transferrin?
- Are the heritability estimates significant?
- Are the heritability values what you expect?

Let's use the GRM matrix to estimate the proportion of phenotypic variance explained by additive genome-wide SNPs for height and serum transferrin. Open the terminal or command prompt and execute the following command.

```
./gcta64  --grm Data/PartE/QIMRX_no_twin  --pheno Data/PartE/HT_T_X.pheno \
          --mpheno 1 \
          --reml --out Results/QIMRX_1


./gcta64  --grm data/PartE/QIMRX_no_twin  --pheno data/PartE/HT_T_X.pheno \
          --mpheno 2 \
          --reml --out Results/QIMRX_2
```

Listing 4: *Estimating variance components via GREML*

We will use R to take a look at the output files that GCTA has calculated. Follow the listing below to read in the files and answer the following questions.

```
> # Read in GREML result files
>
> hsq.1 <- read.table("Results/QIMRX_1.hsq", header = TRUE, fill = TRUE)
> hsq.2 <- read.table("Results/QIMRX_2.hsq", header = TRUE, fill = TRUE)
> head(hsq.1)
```

```
    Source      Variance       SE
1     V(G)      0.637715 0.110157
2     V(e)      0.384206 0.104987
3       Vp      1.021921 0.027815
4 V(G)/Vp      0.624036 0.103832
5     logL -1400.300000       NA
6    logL0 -1418.770000       NA
```

If you prefer the command line you can do this in one line at the command line

```
$ # Look at the heritability file
$ cat Results/QIMRX_2.hsq
Source  Variance  SE
V(G)  0.475748  0.107238
V(e)  0.472528  0.103655
Vp  0.948275  0.027997
V(G)/Vp 0.501698  0.110224
logL  -1098.650
logL0 -1109.236
LRT 21.171
df  1
Pval  2e-06
n 2334
```

Listing 5: *Command line GCTA .hsq file*

3 Unrelated individuals

We will now take a closer look at some of the properties of the GRM using R. You should already have the GRM read in.

```
> # Name the columns of the GRM
> names(grm) <- c("IND_1", "IND_2", "SNP_NUM", "REL")
> dim(grm)
```

```
[1] 4768 4768
```

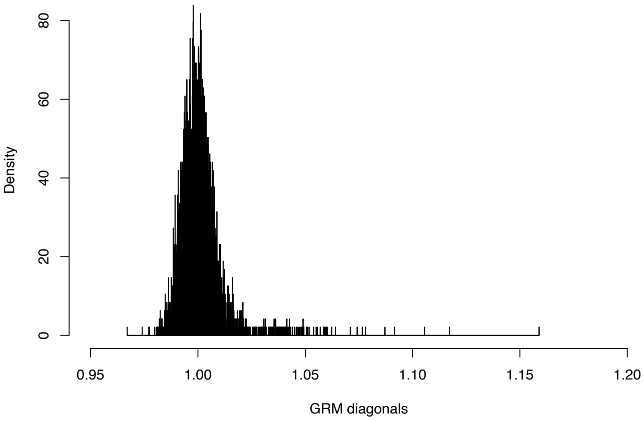
```
> # Take out the diagonal elements
> grm.diag <- diag(grm)
> length(grm.diag)
```

```
[1] 4768
```

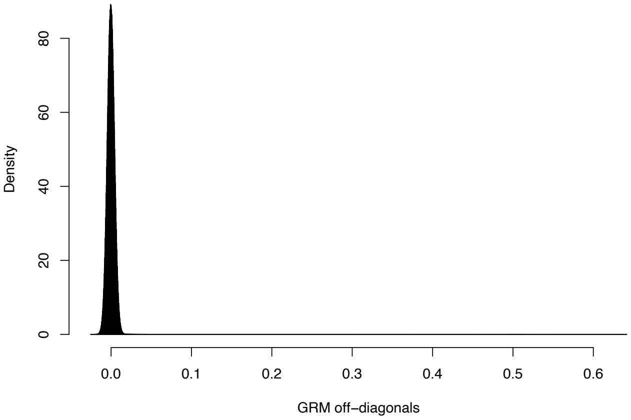
```
> head(grm.diag)
```

```
      883       884       885       886       887       888 
0.9824743 0.9971558 1.0000379 1.0015680 1.0079734 1.0066637 
```

```
> # Take out the GRM off-diagonal elements
> grm.off.diag <- grm[upper.tri(grm)]
> # Make a histogram of the diagonals
> hist(grm.diag, breaks = 2500, freq = F,
+      xlab = "GRM diagonals", xlim = c(0.95, 1.2), main = "")
```



```
> # Make a histogram of the GRM off-diagonal relatedness estimates
> par(mfrow = c(1, 1))
> hist(grm.off.diag, breaks = 2500, freq = F,
+       xlab = "GRM off-diagonals", main = "")
> hist(grm.off.diag[which(grm.off.diag > 0.1)],
+      breaks = 200, freq = F,
+      xlab = "GRM off-diagonals", xlim = c(0.1, 1.1), main = "")
```



In the previous results, the relatedness between individuals may have affected the estimate of heritability (previous figure panel 2). We will remove some individuals to reduce the maximum relatedness coefficient and see whether the Results change. This is done with `GCTA` via the `-grm-cutoff 0.05` flag. This will create a new GRM which you will need to use in the heritability estimation process in Exercise 3. Execute the following command in the terminal.

```
% Removing relatedness
./gcta64  --grm Data/PartE/QIMRX_no_twin  --grm-cutoff 0.05 \
          --make-grm --out Results/QIMRX_nr
```

Listing 6: *Looking at GRM diagonals and off-diagonals*

Exercise 2.

- Repeat the REML analyses as in Exercise 1 but with relatedness removed i.e., just change the GRM to `QIMRX_nr`
- Compare the Results with those in Exercise 1 and from Yang et al. [2010] (for trait 1 height)

```
> # Read in GREML result files without relatedness.
> grm.nr <- read_GRMBin("Results/QIMRX_nr.grm")
> # Create the same figures as above
> # Note that these are only example file extensions
> # and may change depending on what you want to call the files
> hsq.1.nr <- read.table("Results/QIMRX_1_nr.hsq",
+                        header = TRUE, fill = TRUE)
> hsq.2.nr <- read.table("Results/QIMRX_2_nr.hsq",
+                        header = TRUE, fill = TRUE)
> hsq.1.nr
```

```
     Source      Variance       SE
1      V(G)  4.179510e-01 0.139027
2      V(e)  6.066030e-01 0.137142
3        Vp  1.024554e+00 0.029773
4   V(G)/Vp  4.079340e-01 0.134190
5      logL -1.221135e+03       NA
```

```
6     logL0 -1.226198e+03       NA
7       LRT  1.012700e+01       NA
8        df  1.000000e+00       NA
9      Pval  7.306500e-04       NA
10        n  2.386000e+03       NA
```

```
> hsq.2.nr
```

```
     Source     Variance       SE
1      V(G)  4.57216e-01 0.147237
2      V(e)  4.75089e-01 0.144697
3        Vp  9.32304e-01 0.029850
4   V(G)/Vp  4.90415e-01 0.155778
5      logL -9.13756e+02       NA
6     logL0 -9.19545e+02       NA
7       LRT  1.15790e+01       NA
8        df  1.00000e+00       NA
9      Pval  3.33420e-04       NA
10        n  1.96700e+03       NA
```

4 Haseman-Elston regression

We are now going to conduct a Haseman-Elston regression. This analysis can also be conducted using GCTA <http:gcta.freeforums.net/thread/174/hereg-haseman-elston-regression-analysis/>. However, we are going to conduct the analysis in R, which will help to visualise exactly what is being computed. We have preprepared a file `he.grm.txt` file that has one column for the product of the phenotypes of all pairs of individuals with height phenotypes, and another column containing the estimated relatedness calculated from SNP markers for those pairs. Run through the code line-by-line.

```
> #install.packages("readr")
> library("readr")
> he.grm <- read_delim("Data/PartE/he.grm.txt", col_names = F, delim = "\t")
> names(he.grm) <- c("ID1", "ID2", "SNPs", "REL","PROD")
> head(he.grm)
> plot(he.grm$REL, he.grm$PROD)
> # regress the phenotypic product on the relatedness
> lm.he.1 <- lm(he.grm$PROD ~  he.grm$REL)
> summary(lm.he.1)
> # Remove relatedness > 0.025
> he.grm.nr <- he.grm[-which(he.grm$REL > 0.025), ]
> plot(he.grm.nr$REL, he.grm.nr$PROD)
> # Regress the phenotypic product on the relatedness
> lm.he.2 <- lm(he.grm.nr$PROD ~  he.grm.nr$REL)
> summary(lm.he.2)
```