Genome-wide Association Studies

Practical 2: Running a GWAS!

Genetics & Genomics Winter School
Alesha Hatton
8/7/2025

Different ways of running GWAS

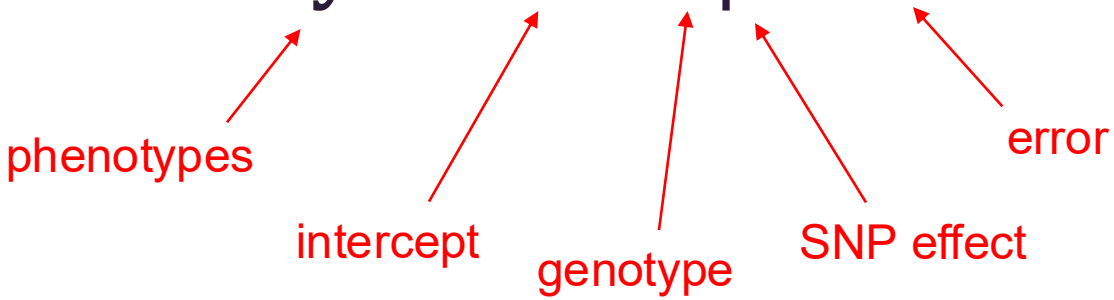
- Running a GWAS in unrelated individuals using PLINK (+/- covariates)
  - Quantitative trait
  - Binary trait
- Including relatives using GCTA

- Look at output, generate Manhattan plots, qq-plots & calculate λ_{GC}

Unrelated individuals with a quantitative trait in PLINK

Model:

$$\mathbf{y} = \mathbf{1}\alpha + \mathbf{x}\beta + \mathbf{e}$$



In PLINK:

plink --bfile <geno file> --assoc --pheno <pheno file>

Unrelated quantitative trait in PLINK

```
[alhatto@ws01 ~]$ plink --bfile /data/module1/5_GWASPrac/theSimsQC --assoc --pheno
/data/module1/5_GWASPrac/BMI.pheno --out raw
PLINK v1.90b7 64-bit (16 Jan 2023)             www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to raw.log.
Options in effect:
  --assoc
  --bfile /data/module1/5_GWASPrac/theSimsQC
  --out raw
  --pheno /data/module1/5_GWASPrac/BMI.pheno

128291 MB RAM detected; reserving 64145 MB for main workspace.
298697 variants loaded from .bim file.
9321 people (4986 males, 4335 females) loaded from .fam.
9321 phenotype values present after --pheno.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 9321 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.99775.
298697 variants and 9321 people pass filters and QC.
Phenotype data is quantitative.
Writing QT --assoc report to raw.qassoc ... done.
```

Output, quantitative trait

```
head raw.qassoc
 CHR          SNP         BP    NMISS       BETA         SE          R2         T          P
   1    rs3131972     752721     9315    -0.2183     0.3057   5.475e-05   -0.7141     0.4752
   1    rs4246503     884815     9314    -0.2145     0.5204   1.824e-05   -0.4121     0.6803
   1    rs3748594     886384     9311     -1.072      0.581   0.0003653    -1.844    0.06515
   1   rs28504611     908414     9313    -0.2278     0.7016   1.132e-05   -0.3247     0.7454
   1    rs2341354     918573     9308    -0.2993     0.2276   0.0001858    -1.315     0.1885
   1    rs2341362     927309     8969     -0.131     0.5609   6.087e-06   -0.2336     0.8153
   1      rs15842     948921     9309    -0.0564      0.549   1.134e-06   -0.1027     0.9182
   1   rs13303287     987670     9308   -0.02286     0.4465   2.816e-07   -0.0512     0.9592
   1    rs3934834    1005806     9315    -0.3065     0.2998   0.0001122    -1.022     0.3067
```

SNP effect

standard error

R^2 (variance explained)

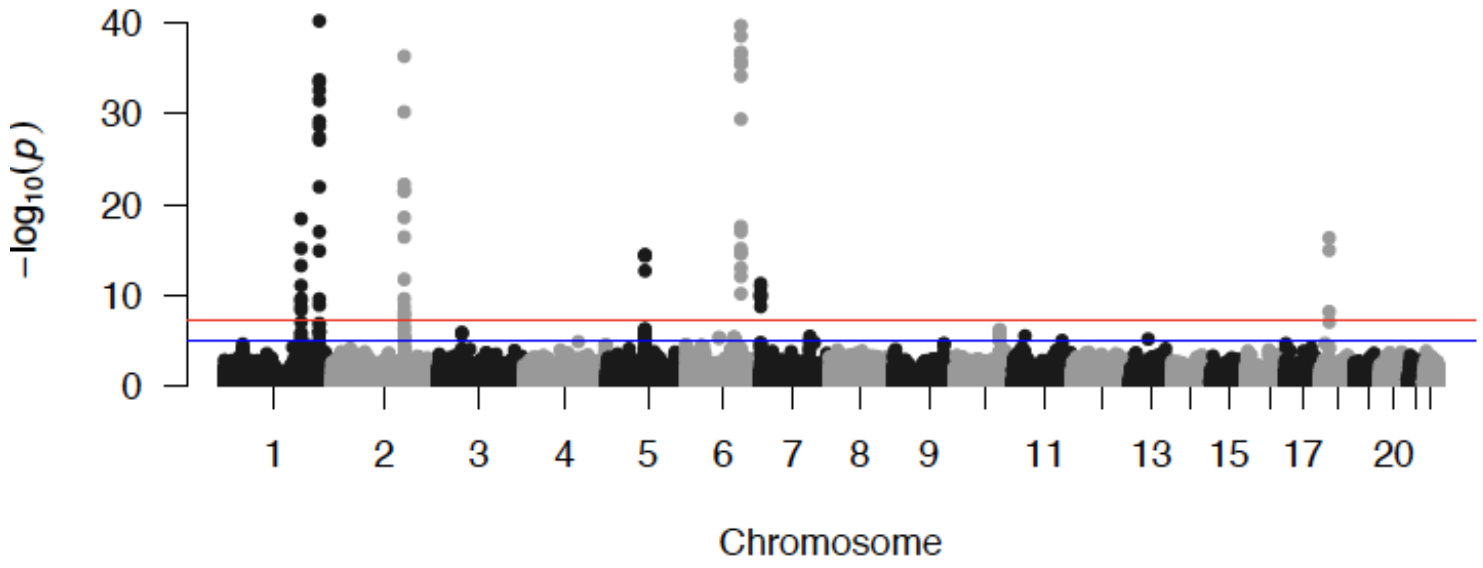
T-test statistic (beta/se)

P-value

Manhattan plot

Use R

library(qqman)
d = read.table(“plink.qassoc”, head=T)
manhattan(d)



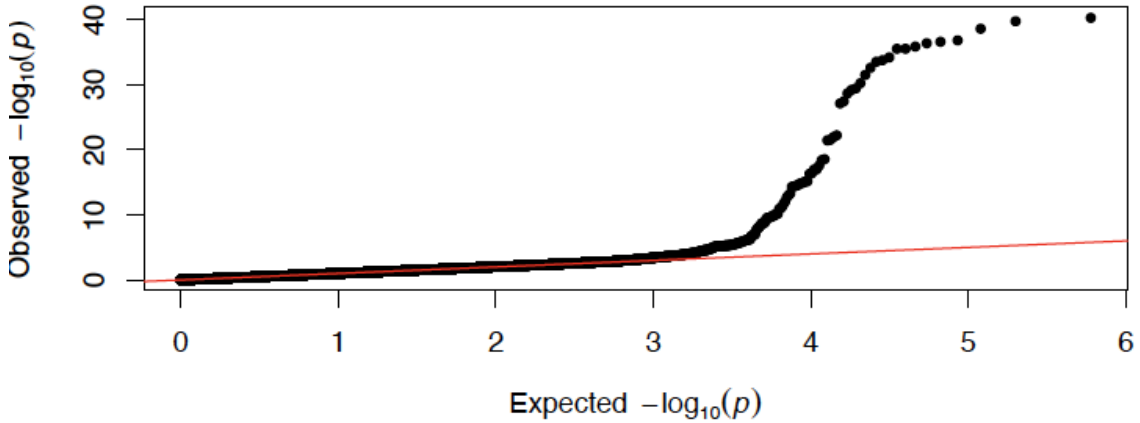
QQ plot & genomic inflation factor (λ_{GC})

- **QQ-plot** is visual approach for comparing 2 distributions

  in our case the expected & observed pvalues from chi-squared distribution

  - i.e. does my test statistic deviate from the null?

```
library(qqman)
d = read.table("plink.qassoc", head=T)
qq(d$P)
```



- **Genomic inflation factor** - expected value of 1.0

$$\lambda = \frac{\text{median of observed } \chi^2 \text{ statistics}}{\text{median of expected } \chi^2 \text{ statistics under the null}}$$

qchisq(1-median(d$P),1)/qchisq(0.5,1)

Multiple testing

Human genetics – Genome-wide significance threshold of 5×10^{-8}

Outside of human genetics, its often unclear what p-value threshold (α) to use. Two options:

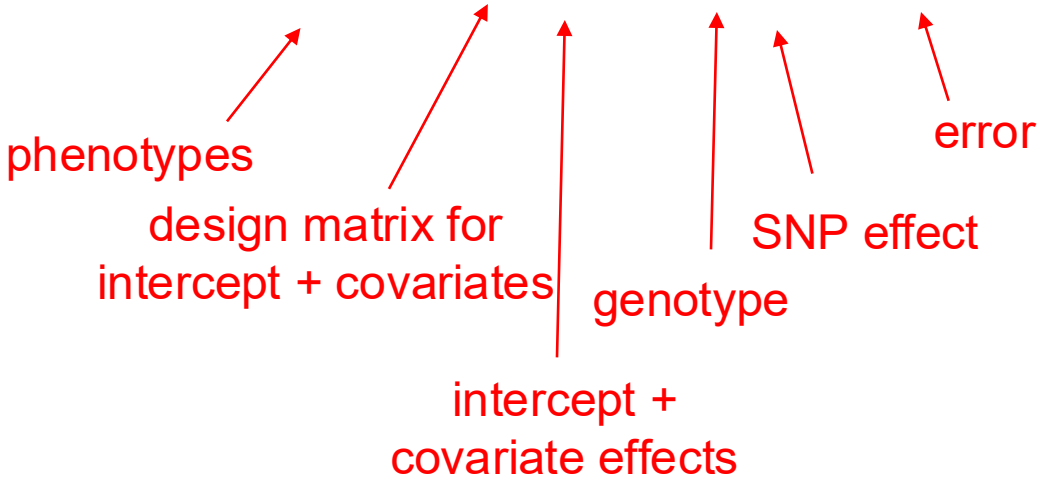
- **False-discovery rate (FDR)**, useful to gage how many false-positive you expect in your results.
  - If we test 1M loci with $\alpha = 1 \times 10^{-5}$, we expect $1 \times 10^{6} \times 1 \times 10^{-5} = 10$ sig. loci by chance
  - Say we observe 150 sig. loci

  FDR = expected/observed = 10/150 = 0.067 or 6.7%
- **Bonferroni correction**, sometimes used but often too stringent as it assumes independent tests.
  - If we test 1M loci and we want $\alpha = 0.01$, then adjusted P-value threshold = $0.01/1 \times 10^{6} = 1 \times 10^{-8}$

Unrelated quantitative trait in PLINK with covariates

Model:

$$\mathbf{y} = \mathbf{W}\boldsymbol{\alpha} + \mathbf{x}\beta + \mathbf{e}$$



In PLINK:

plink --bfile <geno file> --linear --covar <covar file > --pheno <pheno file>

Alternatives: regress the phenotype against the covariates in R and create a new phenotype file with the residuals OR use --fastGWA-lr with --covar in GCTA

Binary trait in PLINK

To perform a standard case/control association analysis, use the option:

```
plink --file mydata --assoc
```

which generates a file

```
plink.assoc
```

which contains the fields:

```
CHR     Chromosome
SNP     SNP ID
BP      Physical position (base-pair)
A1      Minor allele name (based on whole sample)
F_A     Frequency of this allele in cases
F_U     Frequency of this allele in controls
A2      Major allele name
CHISQ   Basic allelic test chi-square (1df)
P       Asymptotic p-value for this test
OR      Estimated odds ratio (for A1, i.e. A2 is reference)
```

Alleles

	1	2	Total
Case	n_1	n_2	2N
Ctrl	m_1	m_2	2M
Total	T_1	T_2	2(N+M)

2x2 contingency table

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Binary trait in PLINK

Phenotype coding: 1 control, 2 case

- --1 flag for data coded as 0 control, 1 case

Use logistic regression if need to correct for covariates

plink --bfile <geno file> --logistic --covar <covar file > --pheno <pheno file>

Be careful of case-control imbalance! >> Inflate type I error rate

Binary trait in UKBB	N_{Case}	$N_{Control}$
Colorectal cancer	4,562	382,756

GWAS with relatives

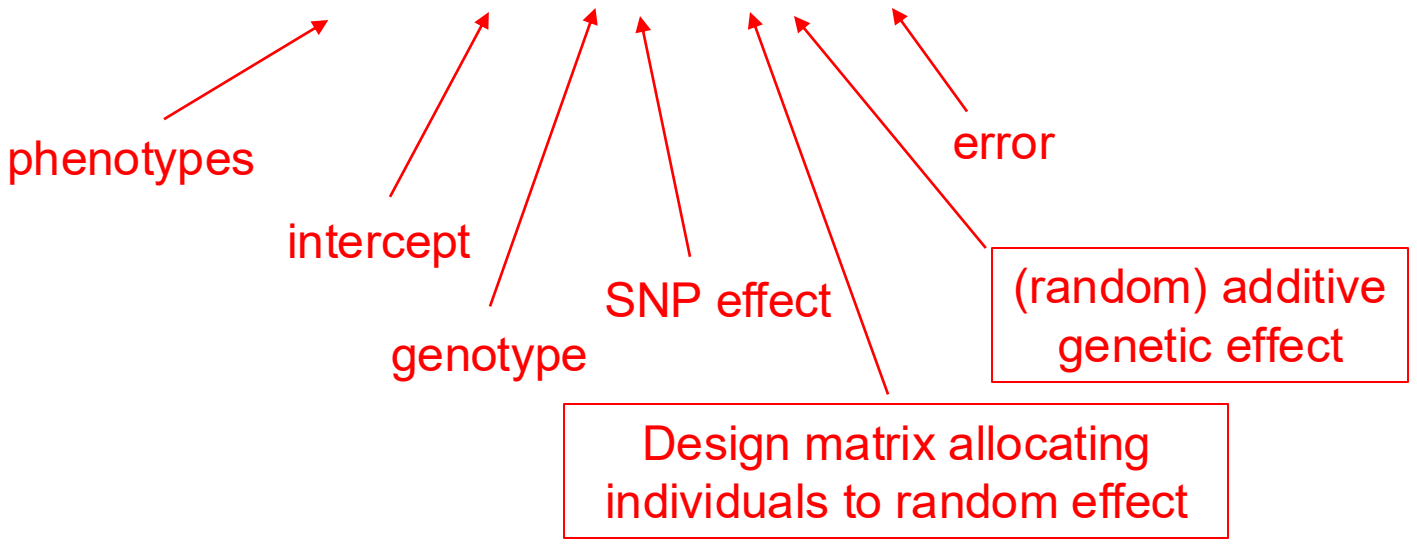
What if we have lots of close relatives ($\pi > 0.05$) - we lose too many individuals if we perform relatedness filtering

There are models that account for this related structure

We can use the -- fastGWA-mlm and --grm-sparse flags in GCTA to fit a sparse genomic relationship matrix (GRM) to model the covariance between closely related individuals

Model:

$$\mathbf{y} = \mathbf{1}\alpha + \mathbf{x}\beta + \mathbf{Z}\mathbf{g} + \mathbf{e}$$



Step 1 - making GRM

Use GCTA at the command line with the -–make-grm-bin flag, e.g.

```
gcta64 --bfile data --make-grm-bin data2 --out data_grm
```

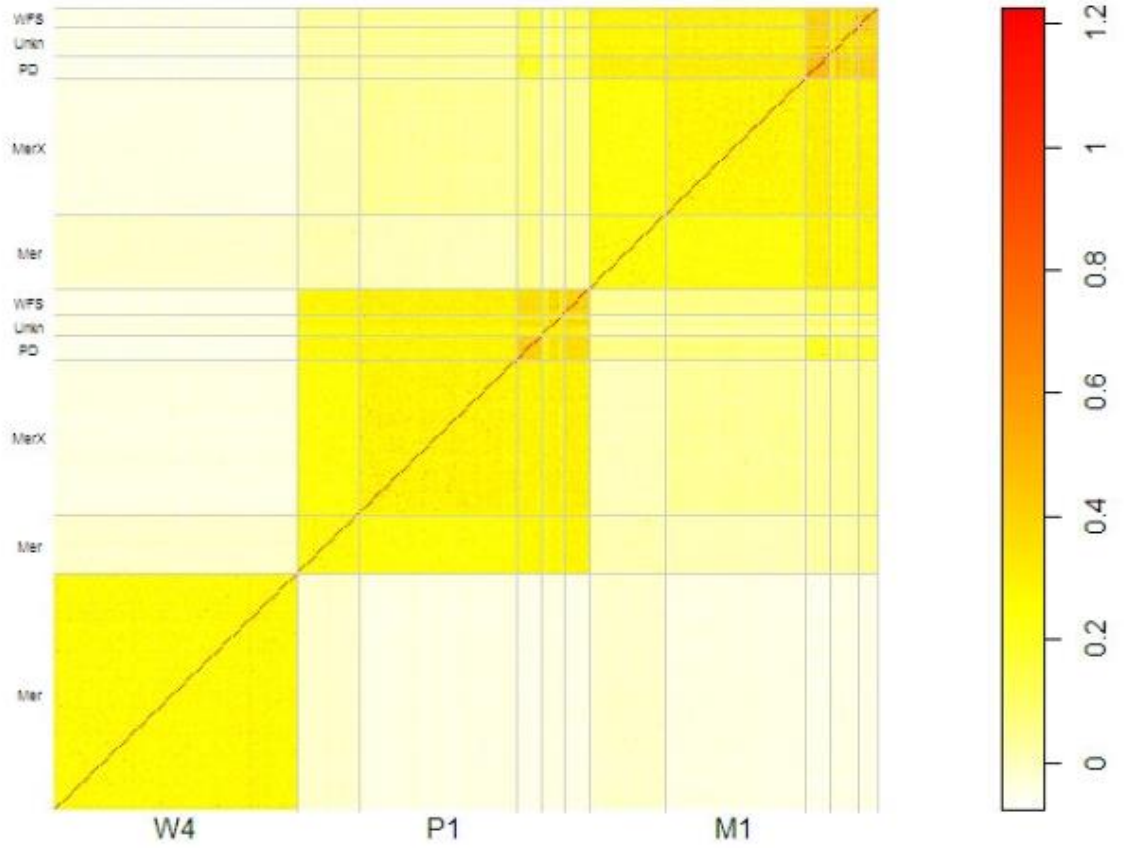
Files produced:

- data_grm.grm.bin → binary file with lower triangle elements of GRM
- data_grm.grm.N.bin → binary file with number of SNPs in GRM
- data_grm.id → list of IDs corresponding to GRM order

Step 1 - making GRM

- Square matrix
- Off-diagonal elements of the GRM estimate the genomic relationship (π) between pairs [i.e. average allele sharing]
- Diagonal has mean 1
- In human genetics, ‘close relatives’ are pairs with π > 0.05

Example GRM from sheep with ½ sib families



Kemper et al. (2011) *Genetics Research*

Step 2 - making a sparse GRM

Use GCTA at the command line with the -–make-bK-sparse flag

This will **set GRM values < 0.05 to zero**

gcta64 --grm data2 -–make-bK-sparse 0.05 --out data2_sparse

Files produced:

- data2_sparse.grm.sp → index and relationships over 0.05 from GRM
- data2_sparse.grm.id → corresponding ID file

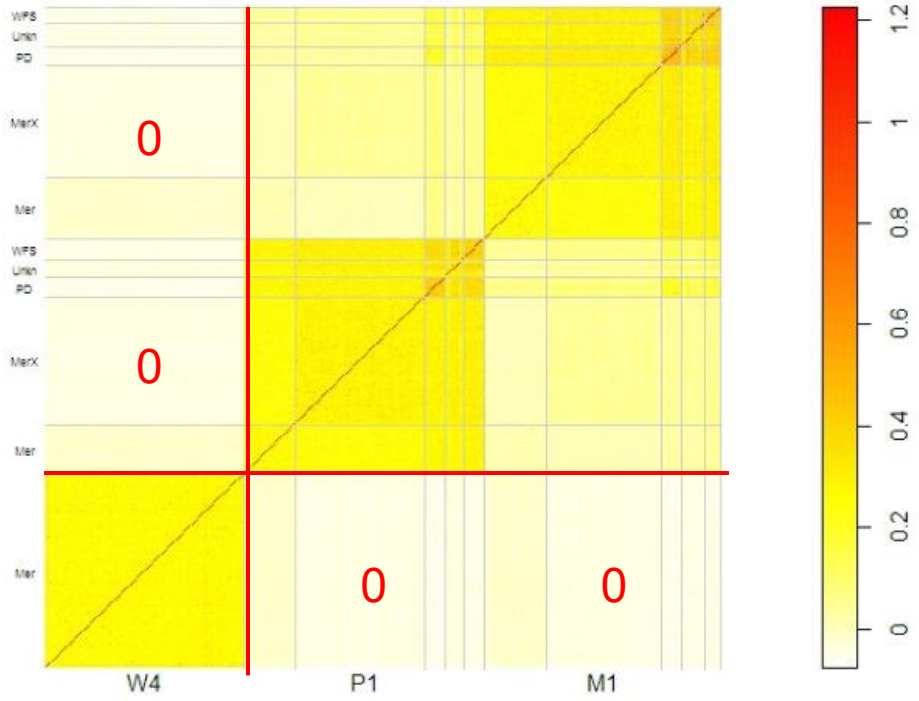
> test_sp_grm.grm.sp (columns are the indexes of a pairs of individuals and the corresponding GRM value)

```
0   0   0.999106
1   1   0.993465
...
```

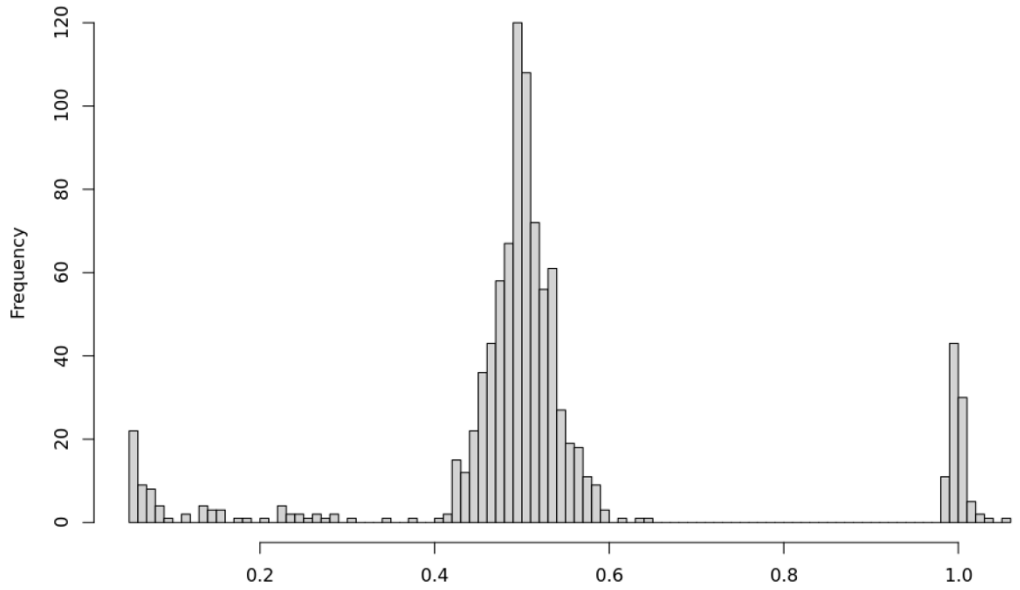
Note: "0" indicates the first individual in the *.grm.id file.

Step 2 - making a sparse GRM

Sheep example



A histogram of the elements in the sparse matrix



Step 3 - running fastGWA

- Use GCTA at the command line with the --fastGWA-mlm and --grm-sparse flag, e.g.

gcta64 --bfile data --fastGWA-mlm --grm-sparse data2_sparse --pheno simData3.phen --out assocSparse

Binary traits --fastGWA-mlm-binary

Covariates --qcovar <file> --covar <file>

Lets get on with the practical..

- Running a GWAS in unrelated individuals using PLINK (+/- covariates)
  - Quantitative trait
  - Binary trait
- Including relatives using GCTA

- Look at output, generate Manhattan plots, qq-plots & calculate λ_{GC}