Genome-wide Association Studies

Practical 2: Do a GWAS!

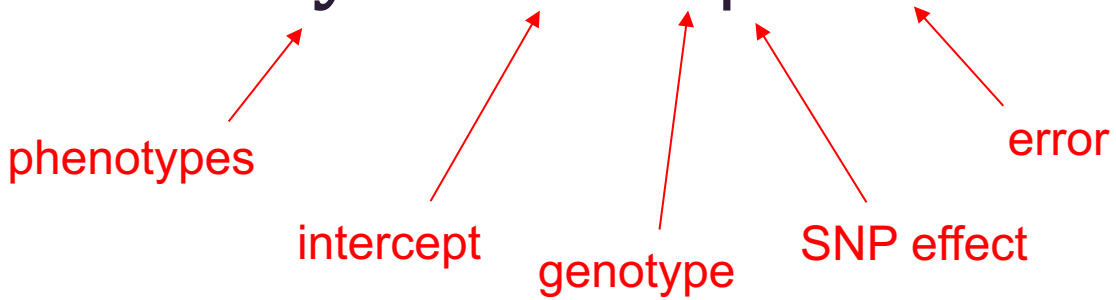
Running a GWAS

- In the practical this afternoon, we will use two programs (PLINK and GCTA) to run a GWAS, with a 3 different ‘flavours’
  - unrelated individuals in PLINK for quantitative trait (+/- covariates)
  - unrelated individuals in PLINK for binary trait (+/- covariates)
  - Including relatives in GCTA for a quantitative trait
- We are assuming here that we are using QC’d genotype & phenotype files
- Look at output, generate Manhattan plots, qq-plots & calculate λ_{GC}

Unrelated 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

```
[allan@analysis1 ~]$ plink --bfile /data/module1/gwas/part2/gwas --assoc --pheno
 /data/module1/gwas/part2/Fasting_Insulin_QC.phen
PLINK v1.90b6.26 64-bit (2 Apr 2022)           www.cog-genomics.org/plink/1.9/
(C) 2005-2022 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to plink.log.
Options in effect:
  --assoc
  --bfile /data/module1/gwas/part2/gwas
  --pheno /data/module1/gwas/part2/Fasting_Insulin_QC.phen

64141 MB RAM detected; reserving 32070 MB for main workspace.
277719 variants loaded from .bim file.
11780 people (5346 males, 6434 females) loaded from .fam.
11770 phenotype values present after --pheno.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 11780 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.995966.
277719 variants and 11780 people pass filters and QC.
Phenotype data is quantitative.
Writing QT --assoc report to plink.qassoc ... done.
```

Output, quantitative trait

```
delta2:~/60days/UQWS_2023/5_unrelGWAS$ head raw.qassoc
 CHR          SNP         BP    NMISS       BETA         SE         R2         T            P
   1   rs12562034     768448    11683     0.2037     0.1314  0.0002056      1.55       0.1212
   1    rs4040617     779322    11667    0.02397     0.1193  3.463e-06     0.201       0.8407
   1    rs4970383     838555    11687    0.03148    0.09247  9.915e-06    0.3404       0.7336
   1     rs950122     846864    11564    0.04572     0.1012  1.767e-05     0.452       0.6513
   1    rs6657440     850780    11687   -0.06427     0.0819  5.271e-05   -0.7848       0.4326
   1   rs13303101     862124    11689    0.09545     0.2875  9.434e-06    0.3321       0.7399
   1    rs1110052     873558    11654   -0.01181    0.09043  1.464e-06   -0.1306       0.8961
   1    rs3748592     880238    11697    -0.1481     0.1775  5.951e-05   -0.8343       0.4041
   1    rs3748593     880390    11696    -0.5318     0.2519   0.000381    -2.111      0.03478
delta2:~/60days/UQWS_2023/5_unrelGWAS$
```

Output, quantitative trait

```
delta2:~/60days/UQWS_2023/5_unrelGWAS$ head raw.qassoc
 CHR          SNP         BP    NMISS       BETA         SE          R2         T          P
   1   rs12562034     768448    11683     0.2037     0.1314   0.0002056      1.55     0.1212
   1    rs4040617     779322    11667    0.02397     0.1193   3.463e-06     0.201     0.8407
   1    rs4970383     838555    11687    0.03148    0.09247   9.915e-06    0.3404     0.7336
   1     rs950122     846864    11564    0.04572     0.1012   1.767e-05     0.452     0.6513
   1    rs6657440     850780    11687   -0.06427     0.0819   5.271e-05   -0.7848     0.4326
   1   rs13303101     862124    11689    0.09545     0.2875   9.434e-06    0.3321     0.7399
   1    rs1110052     873558    11654   -0.01181    0.09043   1.464e-06   -0.1306     0.8961
   1    rs3748592     880238    11697    -0.1481     0.1775   5.951e-05   -0.8343     0.4041
   1    rs3748593     880390    11696    -0.5318     0.2519    0.000381    -2.111    0.03478
```

SNP effect

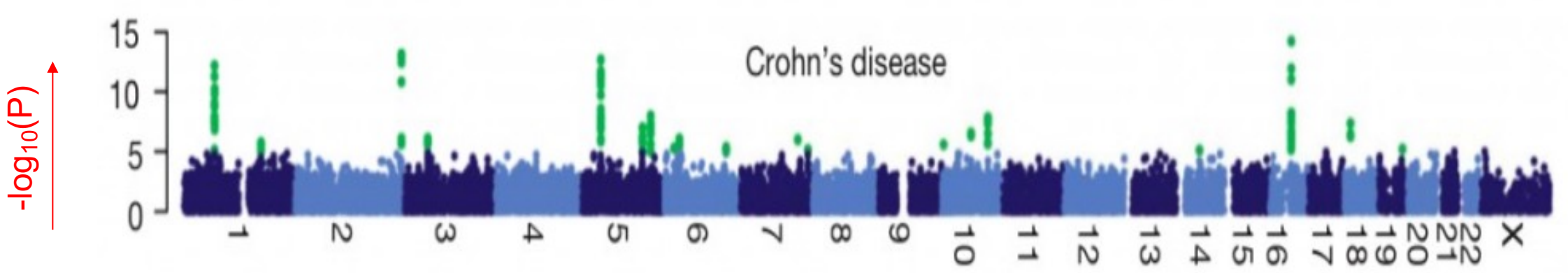
standard error

R^2 (variance explained)

T-test statistic (beta/se)

P-value

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



- Use R

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

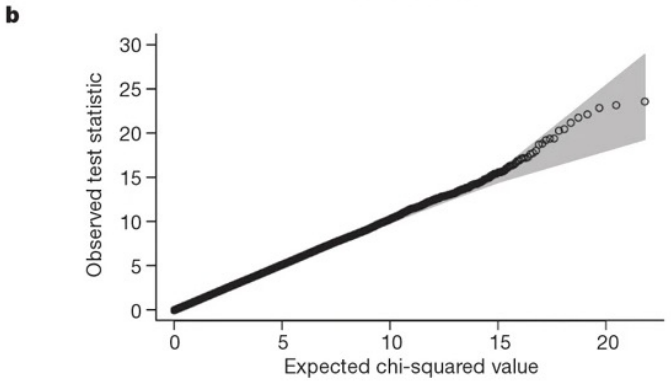
- Genomic inflation factor - expected value of 1.0

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

QQ plot & multiple testing

- QQ-plot is visual approach for comparing 2 distributions, in our case the expected & observed 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)
```



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 α = 0.0001, we expect

  $1x10^6$ *x* 0.0001 = 100 sig. loci by chance

  - Say we observe 150 sig. loci

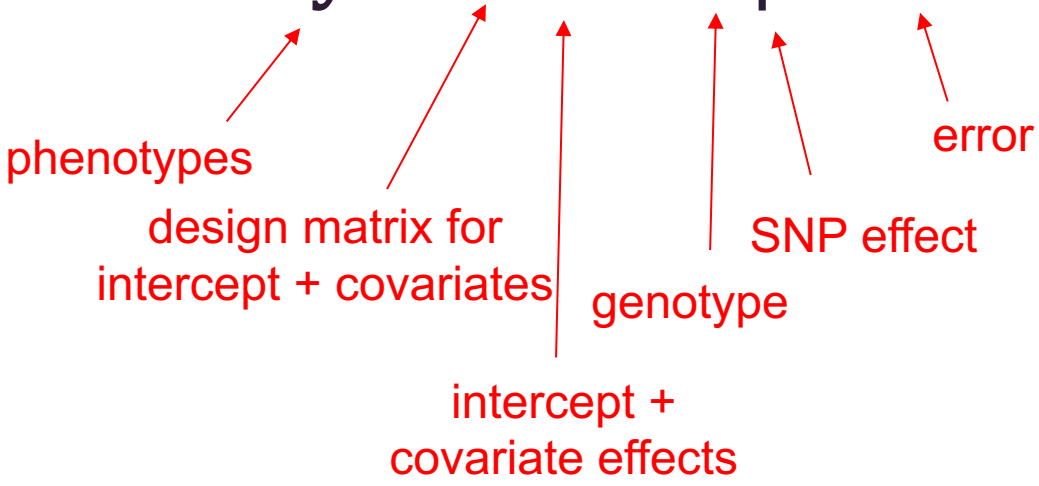
  FDR = expected/observed = 100/150 = 0.67

- Bonferroni correction, sometimes used but often too stringent as it assumes independent tests.
  - If we test 1M loci and we want α = 0.01, then adjusted P-value threshold = $0.01/1x10^6 = 1x10^{-8}$

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

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

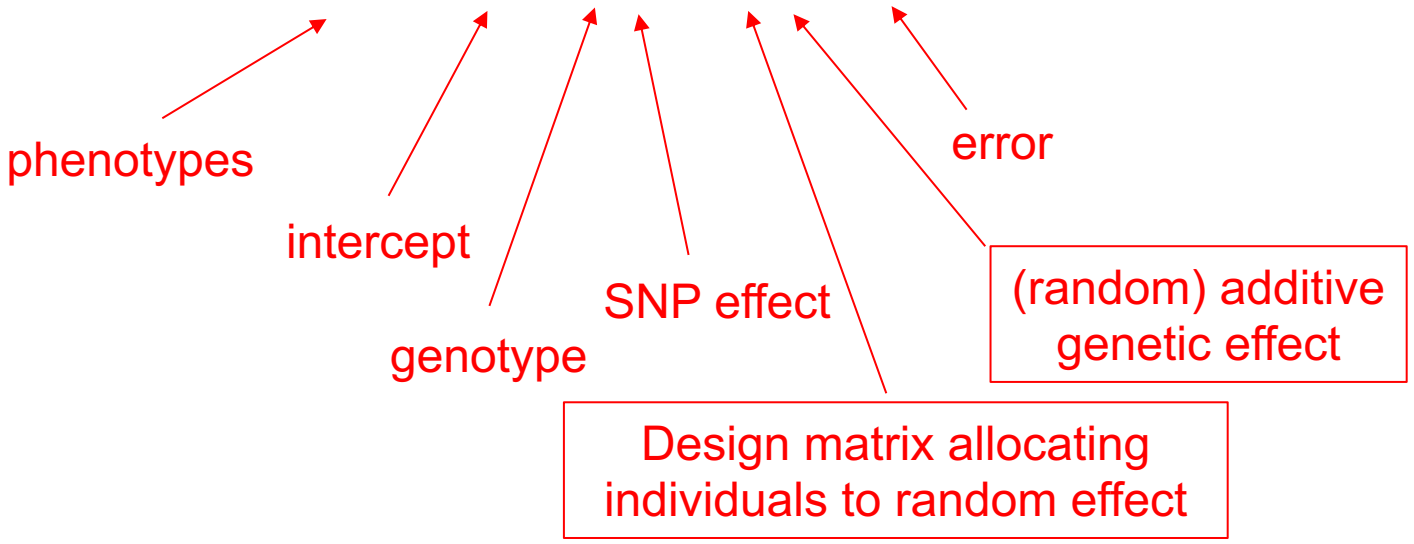
GWAS with relatives

What if we have lots of close relatives ($\pi > 0.05$) and we lose too many records if we remove them all?

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
```

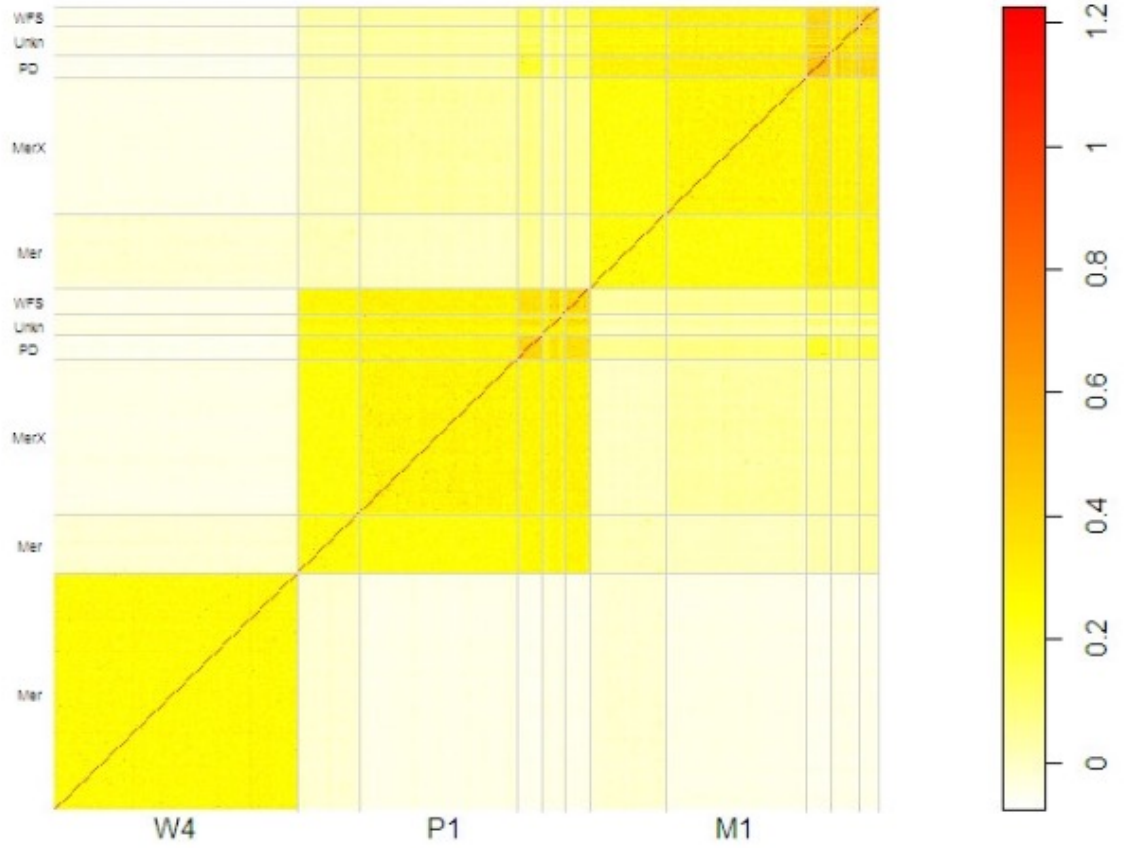
Three files produced:

- 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
- data_grm.log → log file

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
- off-diagonals have mean 0
- 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 to set GRM values < 0.05 to zero, e.g.

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

Three files produced:

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

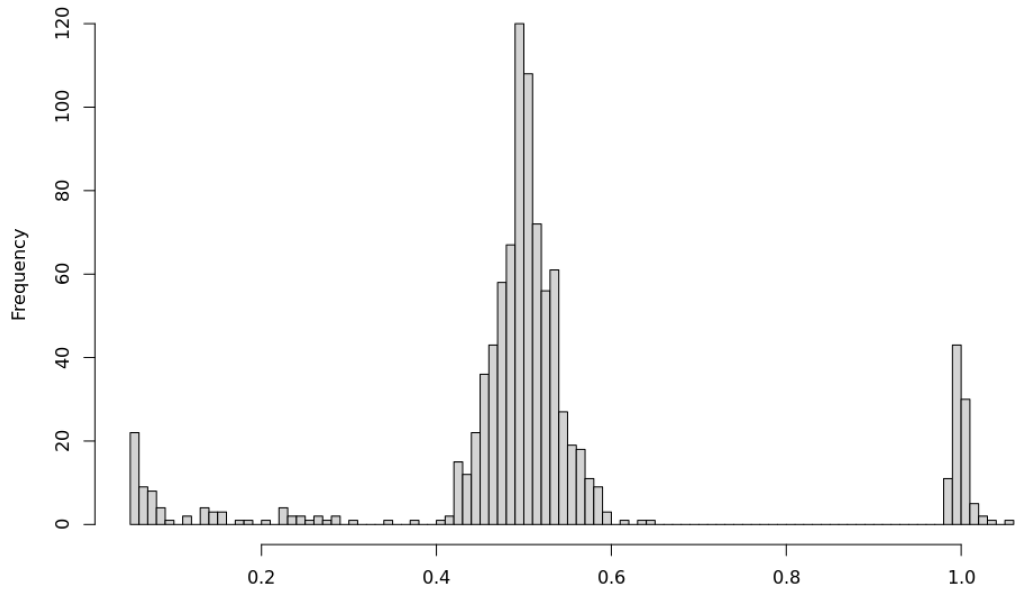
step 2 - making a sparse GRM

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

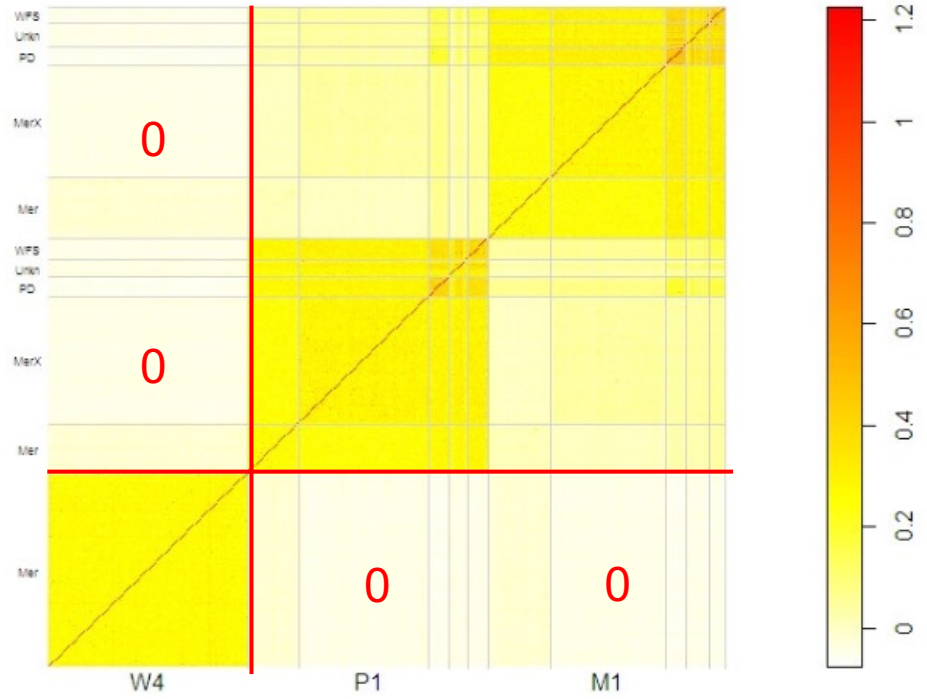
```
0   0   0.999106
1   1   0.993465
...
```

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

A histogram of the elements in the sparse matrix



Sheep example



step 3 - running fastGWA

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

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