2023 International Statistical Genetics Workshop

Day 2. Optional morning GWAS practical.
RUBRIC

March 7, 2023
Luke Evans & Wei Zhou
Practical originally developed by Katrina Grasby and Lucía Colodro Conde

This rubric contains the information for the GWAS practical, as well as additional information on using plink2 to perform association tests and basic functions.

Learning Goals:
1. Run a GWAS using real data
2. Familiarize yourself with the standard types of input and output for a GWAS
3. Practice using the command line
4. Improve your familiarity plink2, a common tool in genetics (beyond GWAS, too)
5. Interpret why peaks form in a Manhattan plot, and how to explore the genome around associated loci

This tutorial aims to make you familiar with genome-wide association analysis, the way in which PLINK2 works (and some of the useful options it provides!), and some tools to visualize your results.

It is important that you explore the outputs and the log files and that you understand what they mean. Please ask if you have any questions, and check the plink file formats page.

Things to note during this practical:
A. There is a plain text file of the instructions alone on the server. You can directly copy and paste from that into the terminal or R. You can also type in the commands directly.
B. There are questions throughout these instructions. These are meant to have you look through the files and output, and think about the commands, what they're doing and how they're operating.
    Commands in plink are highlighted in blue
    Answers to questions and descriptions of the commands in the rubric are highlighted in green
C. Most of the options that you will be using work the same for logistic and linear regression.
D. *Some of the commands below are missing (indicated with XXX).* You will have to determine the correct syntax or command to run the analyses, and discuss the questions proposed.
E. You will find the commands and their output in Day2_GWAS_Practical_RUBRIC.pdf.
    Feel free to take a look through it during the practical, but a good strategy is to
        1. try the commands yourself or with your group
        2. troubleshoot things yourself and ask your group
        3. check the rubric
        4. ask others & faculty
    The rubric has all the expected answers and figures, but the process of doing the GWAS and exploring the output is what will help you learn.

DAY 2. GWAS TUTORIAL

Key information about PLINK2 can be found here:
https://www.cog-genomics.org/plink/2.0/
https://www.cog-genomics.org/plink/2.0/assoc
https://www.cog-genomics.org/plink/2.0/formats

FIRST: Copy the files to your working directory & navigate to that directory:

```
cp -r /faculty/luke/2023/gwas2/ ./
cd gwas2/
```

EXERCISE 1. LOGISTIC REGRESSION (BINARY TRAIT).

1.0. Go to the case-control folder and check the files you have there. The phenotype is Alzheimer's Disease (AD) status.
Source: https://med.miami.edu/faculty/amanda-myers-phd &
https://xzmxbgsv808roffneicreq.on.drv.tw/www.lfun/LFUN/LFUN/DATA.html

```
cd casecontrol
```

1.1. Run a logistic regression for the phenotype (AD status) including the principal components in file adpc.txt as covariates to correct for genetic ancestry.

- [hint: you can find the flags and modifiers that you can include in the logistic and linear regressions on Association analysis > Regression with covariates, https://www.cog-genomics.org/plink/2.0/assoc#glm in the PLINK 2.0 website]

```
plink2 --bfile adclean.cc --glm sex --covar adpc.txt --out 1.1_logistic.ad.cc --threads 4
```

--bfile <prefix> is the dataset filename you will use
--glm is the main plink2 tool to perform associations. Including the 'sex' modifier includes sex (which is in the fam file) as a covariate
--covar <filename> specifies the file with your covariates to use
--threads <max> This sets the number of threads that the program will use. Generally, using more threads makes things faster.

Association analysis

Linear and logistic/Firth regression with covariates

```
--glm ['zs'] ['omit-ref'] [{sex | no-x-sex}] ['log10'] ['pheno-ids']
      [{genotypic | hethom | dominant | recessive | hetonly}] ['interaction']
      ['hide-covar'] ['skip-invalid-pheno'] ['allow-no-covars']
      ['single-prec-cc'] [{intercept | cc-residualize | firth-residualize}]
      [{no-firth | firth-fallback | firth}] ['cols='<col set desc.>]
      ['local-covar='<file>] ['local-psam='<file>]
      ['local-pos-cols='<key col #s> | 'local-pvar='<file>] ['local-haps']
      ['local-omit-last' | 'local-cats='<cat. ct> | 'local-cats0='<cat. ct>]
  (aliases: --linear, --logistic)
--ci <size>
--condition <variant ID> [{dominant | recessive}] ['multiallelic']
--condition-list <variant ID file> [{dominant | recessive}] ['multiallelic']
--parameters <number(s)/range(s)...>
--tests ['all'] [number(s)/range(s)...]
--vif <max VIF>
--max-corr <val>
```

--glm is PLINK 2.0's primary association analysis command.

For quantitative phenotypes, --glm fits the linear model

$$\mathbf{y} = \mathbf{G}\beta_G + \mathbf{X}\beta_X + \mathbf{e}$$

for every variant (one at a time), where **y** is the phenotype vector, **G** is the genotype/dosage matrix for the current variant, **X** is the fixed-covariate matrix, and **e** is the error term subject to least-squares minimization. (Dosages are always used when present; if you want to analyze hardcalled genotypes instead, run "--make-pgen erase-dosage" first.) **X** always contains an all-1 intercept column, along with anything loaded by --covar. Missing-dosage rows are excluded, not mean-imputed.

Check the log file. How many cases and controls were detected? How many covariates?

less -S 1.1_logistic.ad.cc.log

less prints out only the first 10 rows of the file, using the “-S” flag prints it out only one window’s worth at a time.

There were 170 cases, 182 controls, 4 covariates detected in the covar file (not including sex, from the fam file)

```
PLINK v2.00a4LM AVX2 (9 Jan 2023)
Options in effect:
  --bfile adclean.cc
  --covar adpc.txt
  --glm sex
  --out 1.1_logistic.ad.cc
  --threads 4

Hostname: ip-10-0-201-238
Working directory: /home/luke/2023/gwas2/casecontrol
Start time: Sun Mar  5 15:56:49 2023

Random number seed: 1678057009
15657 MiB RAM detected; reserving 7828 MiB for main workspace.
Using up to 4 compute threads.
352 samples (164 females, 188 males; 352 founders) loaded from adclean.cc.fam.
297237 variants loaded from adclean.cc.bim.
1 binary phenotype loaded (170 cases, 182 controls).
4 covariates loaded from adpc.txt.
Calculating allele frequencies... done.
--glm logistic-Firth hybrid regression on phenotype 'PHENO1': done.
Results written to 1.1_logistic.ad.cc.PHENO1.glm.logistic.hybrid .

End time: Sun Mar  5 15:57:17 2023
1.1_logistic.ad.cc.log (END)
```

Check the results (stored in 1.1_logistic.ad.cc.PHENO1.glm.logistic.hybrid) and be sure that you understand the content of each of the columns.

How many lines are in the file?

What is the OR of the second variant and the corresponding p-value?

What are the different values in the “TEST” column, and what do they tell you? Which is the one you’re interested in for a GWAS?

[hint: you can check File formats > .glm.logistic, https://www.cog-genomics.org/plink/2.0/formats#glm_logistic, on the PLINK2.0 website].

head 1.1_logistic.ad.cc.PHENO1.glm.logistic.hybrid

head prints out only the first 10 rows of the file

```
luke@ip-10-0-201-238:~/2023/gwas2/casecontrol$ head 1.1_logistic.ad.cc.PHENO1.glm.logistic.hybrid
#CHROM  POS     ID      REF     ALT     A1      FIRTH?  TEST    OBS_CT  OR      LOG(OR)_SE      Z_STAT  P       ERRCODE
1       752566  rs3094315       T       C       C       N       ADD     349     1.35589 0.210165        1.44868 0.147428        .
1       752566  rs3094315       T       C       C       N       PC1     349     1.24498e-48     61.1278 -1.8045 0.0711535       .
1       752566  rs3094315       T       C       C       N       PC2     349     2.6294e+16      59.1399 0.6393  0.522628        .
1       752566  rs3094315       T       C       C       N       PC3     349     1.98209e+21     58.526  0.837892        0.402091        .
1       752566  rs3094315       T       C       C       N       PC4     349     2.93983e-24     54.4944 -0.994299       0.320077        .
1       752566  rs3094315       T       C       C       N       SEX     349     1.17211 0.217716        0.729413        0.465749        .
1       779322  rs4040617       A       G       G       N       ADD     351     1.28062 0.233422        1.05964 0.289311        .
1       779322  rs4040617       A       G       G       N       PC1     351     3.45264e-49     61.142  -1.82506        0.0679926       .
1       779322  rs4040617       A       G       G       N       PC2     351     4.6348e+19      59.0013 0.767487        0.442792        .
```

wc -l 1.1_logistic.ad.cc.PHENO1.glm.logistic.hybrid

counts the number of lines in the file.

This number is much larger than the number of lines in the *bim file used for this analysis. Do you know why? What other information is stored in this file?

Key columns to understand:

- CHROM: Chromosome of the locus
- POS: physical basepair position of the locus
- ID: rsnumber of the variant
- REF: Reference allele
- ALT: Alternate allele
- A1: The allele that is counted for the regression (Not necessarily ALT, but should be the minor allele)
- OR: A1 allele's odds ratio for the case/control phenotype
- SE: Standard error of the allele's beta (beta=ln(OR))
- P: p-value of the test for this locus

.glm.firth, .glm.logistic[.hybrid] (logistic/Firth regression association statistics)

Produced by --glm with a case/control phenotype.

A text file with a header line, and then one line per variant with the following columns:

Header	Column set	Contents
CHROM	**chrom**	Chromosome code
POS	**pos**	Base-pair coordinate
ID	**(required)**	Variant ID
REF	**ref**	Reference allele
ALT1	alt1	Alternate allele 1
ALT	**alt**	All alternate alleles, comma-separated
A1	**(required)**	Counted allele[1] in regression
OMITTED	omitted	Omitted allele
AX	ax	Non-A1 alleles, comma-separated (deprecated)
A1_CT[2]	a1count	Total A1 allele count (can be decimal with dosage data)
ALLELE_CT[2]	totallele	Allele observation count
A1_CASE_CT[2]	a1countcc	A1 count in cases
A1_CTRL_CT[2]	a1countcc	A1 count in controls
CASE_ALLELE_CT[2]	totallelecc	Case allele observation count
CTRL_ALLELE_CT[2]	totallelecc	Control allele observation count
CASE_NON_A1_CT	gcountcc	Case genotypes with 0 copies of A1
CASE_HET_A1_CT	gcountcc	Case genotypes with 1 copy of A1
CASE_HOM_A1_CT	gcountcc	Case genotypes with 2 copies of A1
CTRL_NON_A1_CT	gcountcc	Control genotypes with 0 copies of A1
CTRL_HET_A1_CT	gcountcc	Control genotypes with 1 copy of A1
CTRL_HOM_A1_CT	gcountcc	Control genotypes with 2 copies of A1
A1_FREQ	a1freq	A1 allele frequency
A1_CASE_FREQ	a1freqcc	A1 allele frequency in cases
A1_CTRL_FREQ	a1freqcc	A1 allele frequency in controls
MACH_R2	machr2	MaCH imputation quality metric
FIRTH?	**firth**	Reports whether Firth reg. was used ('firth-fallback' only)
TEST	**test**	Test identifier
OBS_CT	**nobs**	Number of samples in the regression
BETA	beta	Regression coefficient (for A1 allele)
OR	**orbeta**	Odds ratio (for A1 allele)
[LOG(OR)_]SE	**se**	Standard error of log-odds (i.e. beta)
L##	**ci**	Bottom of symmetric approx. confidence interval (with --ci)
U##	**ci**	Top of symmetric approx. confidence interval (with --ci)
Z_[OR_F_]STAT	**tz**	F-statistic for joint test, Wald Z-score for logistic/Firth regression
[LOG10_]P	**p**	Asymptotic p-value (or -log10(p)) for Z/chisq-stat
ERRCODE	**err**	When result is 'NA', an error code describing the reason

For the second SNP: OR=1.28, p=0.29
The effect allele is A1, which is the minor allele, and may or may not be the ALT allele.
So, an OR > 1 means A1 is associated with an increased risk relative to A2, but here, not significantly so (p=0.29).
The "TEST" column tells you the parameter being estimated. "ADD" indicates the additive allelic effect, while PC1-4 indicates the effects of your PC covariates. "SEX" gives the effect of sex (which is in the fam file in this dataset).

Did any SNP associations reach genome-wide significance?

grep ADD 1.1_logistic.ad.cc.PHENO1.glm.logistic.hybrid | awk '$13<5e-8'
or
sort -k13 -g 1.1_logistic.ad.cc.PHENO1.glm.logistic.hybrid | head ### This just sorts it by p-value & prints out the first few lines

What is the grep command doing here?
What does the awk command do?
What is "|" doing?
The grep command searches for lines with "ADD" in them.
What is the "-g" flag doing to the sort command? It sorts it by numerical value, which is convenient when there are letters used (like 5e-8) in that column.
The "|" is the pipe command, which passes the standard output from the command right before it (here, awk) to the standard input of the command after it (here head).
One SNP reached GWS: rs4420638, with p=1.886e-14.
grep searches for a string in each row of a file. Here, it's finding all of the "ADD" strings, which is short for ADDITIVE estimate of the allele's effect.
It's useful because by default PLINK writes out all of the covariate effect estimates as well (the PC1-5, SEX lines), which are good to check, but which aren't what we're most interested in.
So, grep ADD <file> pulls out only the SNP effect estimates.

What if cases and controls had been coded as 1 and 0, respectively? What could have we done to make PLINK interpret this coding appropriately?

[hint: you can check Standard data input > Phenotypes > Phenotype encoding, https://www.cog-genomics.org/plink/2.0/input#pheno, in the PLINK 2.0 website]
Plink uses 1 & 2 for controls & cases, respectively. You can change this with the --1 command

What if sex had not been coded in the fam file, or was in both your covariate and fam files? What could have we done to make PLINK interpret this coding appropriately?

[hint: you can check --glm modifiers in https://www.cog-genomics.org/plink/2.0/assoc#glm, in the PLINK 2.0 website]

You can tell which covariates to use by adding additional commands, such as --covar-col-num or --covar-name.

1.2. Run a logistic regression for the case-control variable AD including the principal components as covariates and hiding the results of the covariates.

plink2 --bfile adclean.cc --glm sex hide-covar --covar adpc.txt --out 1.2_adclean.cc

What's the difference between the sets of results generated in 1.1 and 1.2?

--glm with the modifier “hide-covar” prevents plink from printing out the estimates for each of the covariates for every single locus, saving a lot of file space.

```
luke@ip-10-0-201-238:~/2023/gwas2/casecontrol$ head 1.2_adclean.cc.PHENO1.glm.logistic.hybrid
#CHROM  POS     ID      REF     ALT     A1      FIRTH?  TEST    OBS_CT  OR      LOG(OR)_SE      Z_STAT  P       ERRCODE
1       752566  rs3094315       T       C       C       N       ADD     349     1.35589 0.210165        1.44868 0.147428        .
1       779322  rs4040617       A       G       G       N       ADD     351     1.28062 0.233422        1.05964 0.289311        .
1       1003629 rs4075116       A       G       G       N       ADD     351     1.01761 0.163397        0.106863        0.914898        .
1       1097335 rs9442385       G       T       T       N       ADD     350     0.63612 0.282949        -1.59876        0.109874        .
1       1130727 rs10907175      A       C       C       N       ADD     343     1.36271 0.284606        1.08739 0.276866        .
1       1158631 rs6603781       C       T       T       N       ADD     344     1.51238 0.222741        1.85726 0.0632745       .
1       1165310 rs11260562      C       T       T       N       ADD     352     0.848671        0.373859        -0.438891       0.66074 .
1       1211292 rs6685064       C       T       T       N       ADD     348     0.840127        0.331242        -0.525908       0.598952
1       1268847 rs307378        G       T       T       N       ADD     349     1.19173 0.474992        0.369277        0.711921        .
luke@ip-10-0-201-238:~/2023/gwas2/casecontrol$ {
```

1.3. Run a logistic regression for the case-control variable AD including the principal components as covariates, hiding the results of the covariates, and getting 95% Confidence intervals of odds ratios.

What is a confidence interval?

[hint: you can check again https://www.cog-genomics.org/plink/2.0/assoc#glm, in the PLINK 2.0 website]

plink2 --bfile adclean.cc --glm sex hide-covar --ci 0.95 --covar adpc.txt --out 1.3_adclean.cc --threads 4

--ci 0.95 tells plink to output the 95% CI of the estimates.

```
luke@ip-10-0-201-238:~/2023/gwas2/casecontrol$ head 1.3_adclean.cc.PHENO1.glm.logistic.hybrid
#CHROM  POS     ID      REF     ALT     A1      FIRTH?  TEST    OBS_CT  OR      LOG(OR)_SE      L95     U95     Z_STAT  P       ERRCODE
1       752566  rs3094315       T       C       C       N       ADD     349     1.35589 0.210165        0.898117        2.047   1.44868 0.147428        .
1       779322  rs4040617       A       G       G       N       ADD     351     1.28062 0.233422        0.810457        2.02352 1.05964 0.289311        .
1       1003629 rs4075116       A       G       G       N       ADD     351     1.01761 0.163397        0.738753        1.40174 0.106863        0.914898 .
1       1097335 rs9442385       G       T       T       N       ADD     350     0.63612 0.282949        0.365335        1.10761 -1.59876        0.109874 .
1       1130727 rs10907175      A       C       C       N       ADD     343     1.36271 0.284606        0.780094        2.38046 1.08739 0.276866        .
1       1158631 rs6603781       C       T       T       N       ADD     344     1.51238 0.222741        0.977383        2.34023 1.85726 0.0632745       .
1       1165310 rs11260562      C       T       T       N       ADD     352     0.848671        0.373859        0.407859        1.76591 -0.438891       0.
66074   .
1       1211292 rs6685064       C       T       T       N       ADD     348     0.840127        0.331242        0.438926        1.60804 -0.525908       0.
598952  .
1       1268847 rs307378        G       T       T       N       ADD     349     1.19173 0.474992        0.469745        3.02337 0.369277        0.711921 .
```

If you’re interested in a specific variant and its effect, a CI of the effect size for a variant may be very important to know, and this is a way to get that quickly. You can also always calculate a CI of varying range yourself using a t-distribution.

1.4. Run a logistic regression for the case-control variable AD including the principal components as covariates, hiding the results of the covariates, and getting the allele frequencies.

[hint: you can check Main functions > Basic statistics > --freq or Allele frequency, https://www.cog-genomics.org/plink/1.9/basic_stats#freq, in the PLINK 1.9 website].

plink2 --bfile adclean.cc --glm sex hide-covar --covar adpc.txt --freq --out 1.4_adclean.cc --threads 4

--freq creates a second output, *.afreq, which contains the frequency information.

Explore the output files and note you have an extra one, 1.4_adclean.cc.afrq. [hint: you can check File formats > .afrq in the PLINK website]. In this one, we can see REF, ALT, and ALT_Freq.

```
luke@ip-10-0-201-238:~/2023/gwas2/casecontrol$ head 1.4_adclean.cc.afreq
#CHROM  ID      REF     ALT     ALT_FREQS       OBS_CT
1       rs3094315       T       C       0.161891        698
1       rs4040617       A       G       0.109687        702
1       rs4075116       A       G       0.299145        702
1       rs9442385       G       T       0.0771429       700
1       rs10907175      A       C       0.0758017       686
1       rs6603781       C       T       0.133721        688
1       rs11260562      C       T       0.046875        704
1       rs6685064       C       T       0.058908        696
1       rs307378        G       T       0.0243553       698
```

.acount, .afreq (allele count/frequency report)

Produced by --freq.

A text file with a header line, and then one line per variant with the following columns:

Header	Column set	Contents
CHROM	**chrom**	Chromosome code
POS	pos	Base-pair coordinate
ID	**(required)**	Variant ID
REF	**ref**	Reference allele
ALT1	alt1	Alternate allele 1
ALT	**alt**	All alternate alleles, comma-separated
'REF_FREQ'/'REF_CT'	reffreq	Reference allele frequency/dosage
'ALT1_FREQ'/'ALT1_CT'	alt1freq	Alternate allele 1 frequency/dosage
'ALT_FREQS'/'ALT_CTS'	**altfreq**, alteq, alteqz	Comma-separated freqs/dosages for all alts; 'eq' requests '1=<ALT1 value>,2=<ALT2 value>,...' formatting with zero-values omitted, 'eqz' includes zeroes
'ALT_NUM_{FREQS,CTS}'	altnumeq	Comma-separated freqs/dosages for all alts
'FREQS'/'CTS'	freq, eq, eqz	Comma-separated freqs/dosages for all alleles
'NUM_FREQS'/'NUM_CTS'	numeq	Comma-separated freqs/dosages for all alleles
MACH_R2	machr2	MaCH imputation quality metric
MINIMAC3_R2	minimac3r2	Minimac3 phased-dosage imputation quality metric; **inaccurate unless phased dosages were imported with e.g. "--vcf dosage=HDS"** (dosage=DS is not enough)
OBS_CT	**nobs**	Number of allele observations

EXERCISE 2. LINEAR REGRESSION (CONTINUOUS TRAIT)

2.0. Go to the continuous folder and check the files you have there. The phenotype is a transcript probe (gene expression). Pay attention to the file adclean.cont.txt.

```
cd ../continuous/
head adclean.cont.txt
```

2.1. Run a linear regression for the continuous trait including the genetic principal components as covariates, hiding the results of the covariates, and using the --pheno option. The advantage of using an extra file for phenotypes (and the --pheno option) is that if there were several phenotypes, it would be possible to run analyses on them at the same time (something not possible with ped or fam files). Note that when using the --pheno option, the original ped or fam files can still contain a phenotype in the phenotypc column, but it can be missing (NA/nan/-9). See Standard data input > Phenotypes, https://www.cog-genomics.org/plink/2.0/input#pheno.

```
plink2 --bfile adclean.cont --glm sex hide-covar --pheno adclean.cont.txt --covar adpc.txt --out 2.1_adclean.cont --threads 4
```

What do you notice about the output file name? Because the phenotype file has a column name, it uses that phenotype name in the output by default.

```
luke@ip-10-0-201-238:~/2023/gwas2/continuous$ head 2.1_adclean.cont.GI_34147330-S.glm.linear
#CHROM  POS     ID      REF     ALT     A1      TEST    OBS_CT  BETA    SE      T_STAT  P       ERRCODE
1       752566  rs3094315       T       C       C       ADD     349     -0.00601159     0.0468227       -0.12839        0.897916        .
1       779322  rs4040617       A       G       G       ADD     351     0.0137903       0.0520003       0.265196        0.791017        .
1       1003629 rs4075116       A       G       G       ADD     351     0.0153566       0.0365065       0.420655        0.67427 .
1       1097335 rs9442385       G       T       T       ADD     350     0.0124305       0.0606909       0.204816        0.837837        .
1       1130727 rs10907175      A       C       C       ADD     343     -0.0509092      0.0633483       -0.80364        0.422173        .
1       1158631 rs6603781       C       T       T       ADD     344     -0.035305       0.0490759       -0.719395       0.472396        .
1       1165310 rs11260562      C       T       T       ADD     352     -0.0607221      0.0832358       -0.729519       0.466179        .
1       1211292 rs6685064       C       T       T       ADD     348     -0.0702463      0.0737093       -0.953018       0.341256        .
1       1268847 rs307378        G       T       T       ADD     349     0.0620477       0.105937        0.585704        0.558461        .
```

Phenotypes

```
--pheno ['iid-only'] <filename>
--pheno-col-nums <1-based column number(s)/range(s)...>
--pheno-name <column ID(s)/range(s)...>
--no-psam-pheno
  (aliases: --no-pheno, --no-fam-pheno)
--not-pheno <phenotype ID(s)...>
  (alias: --phenoExcludeList)
```

--pheno causes (additional) phenotype values to be read from the specified space- or tab-delimited file. The first columns of that file must be either FID/IID or just IID (in which case the FID is assumed to be 0). A primary header line is required when using --pheno-name, and optional without it (if it's present, it should begin with 'FID', '#FID', 'IID', or '#IID'). Additional header lines (beginning with '#', not immediately followed by 'FID'/'IID') are permitted before the primary header line. For example:

```
## * If you also need PLINK 1.9 to read this file, add an FID column in front,
##   and fill it with zeroes.
## * 'site' is a categorical variable.  --glm would ignore it if you loaded it
##   as a phenotype (multinomial logistic regression is not implemented), but
##   it's a valid *covariate* for --glm.
#IID  qt1    bmi    site
1110  2.3    22.22  site2
2202  34.12  18.23  site1
...
```

2.2. Run a linear regression for the continuous trait including only PC1 as covariate, hiding the results of the covariate, using the --pheno option.

[hint: again, you can check the commands you can use to run logisic or linear regressions on data input > Covariates, https://www.cog-genomics.org/plink/2.0/input#pheno. For options related to how parameters are handled in other models (e.g., estimating non-additive effects), you can seehttps://www.cog-genomics.org/plink/2.0/assoc

plink2 --bfile adclean.cont --glm hide-covar --pheno adclean.cont.txt --covar adpc.txt --covar-name PC1 --out 2.2_adclean.cont --threads 4

--covar-name <column name> allows you to specify only certain covariates out of your file. Also, this does not include sex as a covariate, because this command doesn't have the –glm modifier 'sex'.

```
luke@ip-10-0-201-238:~/2023/gwas2/continuous$ head 2.2_adclean.cont.GI_34147330-S.glm.linear
#CHROM  POS     ID      REF     ALT     A1      TEST    OBS_CT  BETA    SE      T_STAT  P       ERRCODE
1       752566  rs3094315       T       C       C       ADD     349     -0.0133576      0.0471627       -0.283225       0.777174
1       779322  rs4040617       A       G       G       ADD     351     0.0122987       0.052342        0.234968        0.814372
1       1003629 rs4075116       A       G       G       ADD     351     0.0160724       0.0368213       0.436496        0.662748
1       1097335 rs9442385       G       T       T       ADD     350     0.0105939       0.0613877       0.172573        0.863088
1       1130727 rs10907175      A       C       C       ADD     343     -0.0625219      0.0637446       -0.980819       0.32738 .
1       1158631 rs6603781       C       T       T       ADD     344     -0.0411284      0.0496146       -0.828958       0.407708
1       1165310 rs11260562      C       T       T       ADD     352     -0.0537221      0.0834414       -0.64383        0.520108
1       1211292 rs6685064       C       T       T       ADD     348     -0.0644163      0.0739107       -0.871542       0.384064
1       1268847 rs307378        G       T       T       ADD     349     0.0851028       0.106909        0.796031        0.42656 .
luke@ip-10-0-201-238:~/2023/gwas2/continuous$
```

2.3. Run a linear regression for the continuous trait including the principal components as covariates, hiding the results of the covariates, using the --pheno option, and getting 95% confidence intervals for the beta.

plink2 --bfile adclean.cont --glm sex hide-covar --pheno adclean.cont.txt --covar adpc.txt --ci 0.95 --out 2.3_adclean.cont --threads 4

similar to above, the –ci 0.95 command outputs the 95% CI.

What is different about these results than 2.1 results? How might you use the added information?

If you're interested in a specific variant and its effect, the uncertainty in the effect size may be very important to know, and this is a way to get that quickly. You can also always calculate a CI of varying range yourself using a t-distribution.

```
luke@ip-10-0-201-238:~/2023/gwas2/continuous$ head 2.3_adclean.cont.GI_34147330-S.glm.linear
#CHROM  POS     ID      REF     ALT     A1      TEST    OBS_CT  BETA    SE      L95     U95     T_STAT  P       ERRCODE
1       752566  rs3094315       T       C       C       ADD     349     -0.00601159     0.0468227       -0.0977825      0.0857593       -0.12839        0.897916        .
1       779322  rs4040617       A       G       G       ADD     351     0.0137903       0.0520003       -0.0881284      0.115709        0.265196        0.791017        .
1       1003629 rs4075116       A       G       G       ADD     351     0.0153566       0.0365065       -0.0561948      0.0869081       0.420655        0.67427 .
1       1097335 rs9442385       G       T       T       ADD     350     0.0124305       0.0606909       -0.106522       0.131383        0.204816        0.837837        .
1       1130727 rs10907175      A       C       C       ADD     343     -0.0509092      0.0633483       -0.17507        0.0732511       -0.80364        0.422173        .
1       1158631 rs6603781       C       T       T       ADD     344     -0.035305       0.0490759       -0.131492       0.0608821       -0.719395       0.472396        .
1       1165310 rs11260562      C       T       T       ADD     352     -0.0607221      0.0832358       -0.223861       0.102417        -0.729519       0.466179        .
1       1211292 rs6685064       C       T       T       ADD     348     -0.0702463      0.0737093       -0.214714       0.0742213       -0.953018       0.341256        .
1       1268847 rs307378        G       T       T       ADD     349     0.0620477       0.105937        -0.145585       0.26968 0.585704        0.558461        .
luke@ip-10-0-201-238:~/2023/gwas2/continuous$
```

2.4. Plot the results from 2.3.

We will create three plots to explore the results. For that, we will need at least three columns: chromosome, base pair position, and p-value; having a SNP column (containing the rs number) will allow extra options in our plots. We'll **first create a file containing this information, excluding the markers with no results**.

awk '{print $1,$2,$3,$14}' 2.3_adclean.cont.GI_34147330-S.glm.linear | grep -v NA > plot.adclean.cont.linear.txt

What does the grep command do, and why would you want to use that?

The grep command uses the '-v' modifier so that it searches for lines WITHOUT "NA" in them. These are lines with missing estimates, and we don't want them, so we grab all the lines that don't have "NA" in them instead.

```
luke@ip-10-0-201-238:~/2023/gwas2/continuous$ head plot.adclean.cont.linear.txt
#CHROM POS ID P
1 752566 rs3094315 0.897916
1 779322 rs4040617 0.791017
1 1003629 rs4075116 0.67427
1 1097335 rs9442385 0.837837
1 1130727 rs10907175 0.422173
1 1158631 rs6603781 0.472396
1 1165310 rs11260562 0.466179
1 1211292 rs6685064 0.341256
1 1268847 rs307378 0.558461
```

What's the SNP with the lowest p-value (or top SNP)?

```
sort -k4 -g plot.adclean.cont.linear.txt | head
```

What is the "-g" doing in the line above? Hint: use "man sort" to check the documentation of commands.
The "-g" modifier to the sort command sorts the particular column (4th, specified by -k4), as a number, even though it's got a letter in it sometimes (using notation like 5e-8 as 0.00000005).

```
luke@ip-10-0-201-238:~/2023/gwas2/continuous$ sort -k4 -g plot.adclean.cont.linear.txt | head
#CHROM POS ID P
20 25451180 rs449370 5.07922e-39
20 25397257 rs6050598 5.16834e-39
20 25350325 rs4815412 5.50642e-39
20 25275843 rs2258719 7.66462e-39
20 25291848 rs1888999 7.66462e-39
20 25297909 rs2297497 8.71415e-39
20 25366065 rs6050573 9.24724e-39
20 25433821 rs12428 1.214e-38
20 25270339 rs2257991 1.23612e-38
```

EXERCISE 3. PLOT OUT YOUR RESULTS

You'll next make your own Q-Q and Manhattan plots from these data, then look at the region around the strongest association.

QQ & Manhattan Plots

Open in RStudio (https://workshop.colorado.edu/rstudio/).

Open the script Rscript_qqMan.R (in the folder you copied) and plot the results.

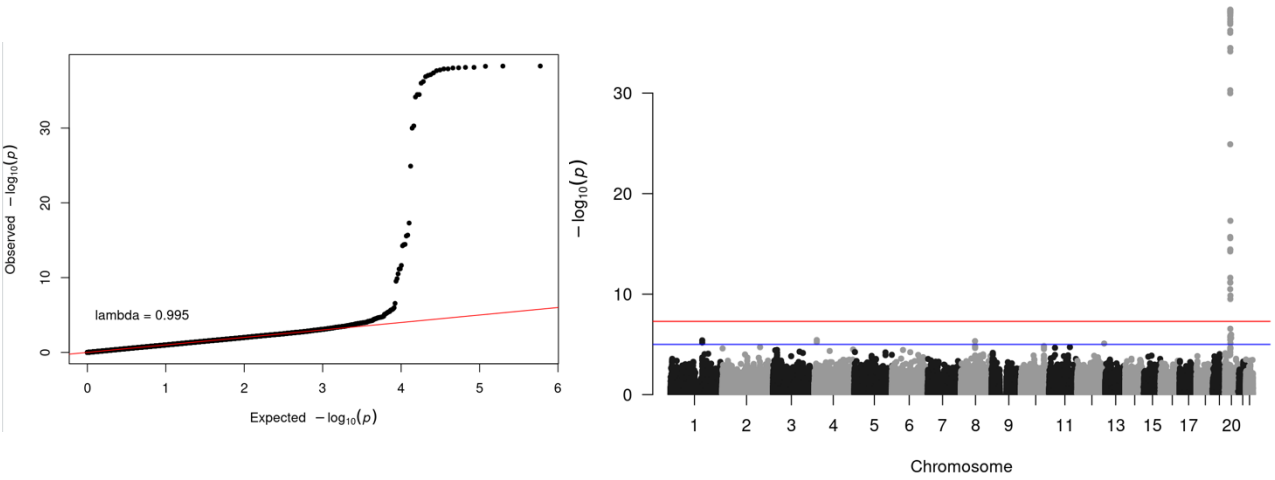
1. Set your working directory to the folder where you have your.
2. Run the script to create both plots.

What is the expected distribution of the p-values? Which plot is comparing the expected to the observed?
Can you locate the strongest association from your work above in both plots?
What is the y-axis of the Manhattan plot and why is it plotted in those units?
What information do you gather from these plots and the lambda value? Do you detect any anomalies?

You need to set your working directory first. This is either by specifying the path in the script or by using Session>Set Working Directory>Choose Directory in the drop-down menu at the top of RStudio.
You also need to tell it which file to use, here, "plot.adclean.cont.linear.text'.
Note that while this file does have a header column, it starts with a "#" which comments out the line and which RStudio doesn't recognize. So, you then have to set the names of the column.
Checking the structure uses str(Data), and will output information for each column in the "Data" object.

```
library(qqman)

# Indicate working directory

setwd("~/2023/gwas2/continuous/") # set your working directory; e.g.~/gwas2/continuous/ or e.
. ~/gwas2/casecontrol/, or use "Session" in the drop-down menu above to set your directory

Data <- read.table('plot.adclean.cont.linear.txt', header=F) # select you file; e.g. plot
.adclean.cc.logistic.txt or plot.adclean.cont.linear.txt
head(Data)

colnames(Data)<-c("CHROM", "POS", "ID", "P") # give it the right column names
head(Data)


# make sure the data is prepared so there is no missing data and the key variables (CHR, BP,
) are numeric.
# How do you check the structure of the data?
str(Data) # What is the command you can use

24:1  (Top Level)                                                    R Sc
```

```
Console  Terminal  Background Jobs
R 4.2.2 · ~/2023/gwas2/continuous/

> colnames(Data)<-c("CHROM", "POS", "ID", "P") # give it the right column names
> head(Data)
  CHROM     POS         ID        P
1     1  752566  rs3094315 0.897916
2     1  779322  rs4040617 0.791017
3     1 1003629  rs4075116 0.674270
4     1 1097335  rs9442385 0.837837
5     1 1130727 rs10907175 0.422173
6     1 1158631  rs6603781 0.472396
> # make sure the data is prepared so there is no missing data and the key variables (CHR, BP, P) a
numeric.
> # How do you check the structure of the data?
> str(Data) # What is the command you can use
'data.frame':   297237 obs. of  4 variables:
 $ CHROM: int  1 1 1 1 1 1 1 1 1 1 ...
 $ POS  : int  752566 779322 1003629 1097335 1130727 1158631 1165310 1211292 1268847 1478153 ...
 $ ID   : chr  "rs3094315" "rs4040617" "rs4075116" "rs9442385" ...
 $ P    : num  0.898 0.791 0.674 0.838 0.422 ...
>
```



Make a regional plot of the strongest association.

1. Still in RStudio, go to the "Files" section (bottom right quadrant of the browser) and check the file plot.adclean.cont.linear.txt and the two jpg images you just created.

2. Go to More > Export. The file will download to your local computer, and you will need to unzip it.

There is also a copy of this same file on the shared files section on the server if you have any trouble exporting this from Rstudio: Day2_plot.adclean.cont.linear.txt

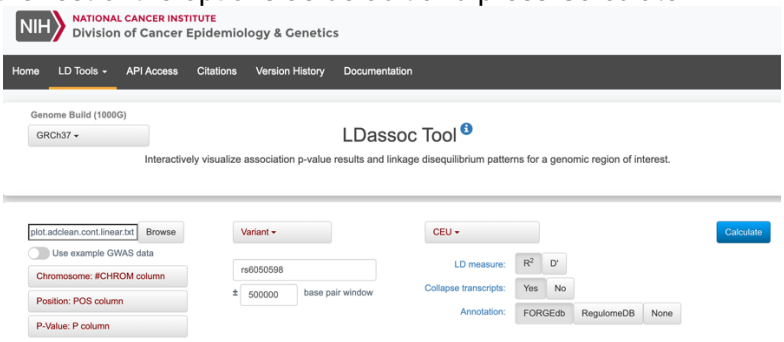
3. Upload your results to LDassoc in LDlink: https://ldlink.nci.nih.gov/?tab=ldassoc.

You can explore association p-value results and LD patterns using this tool.

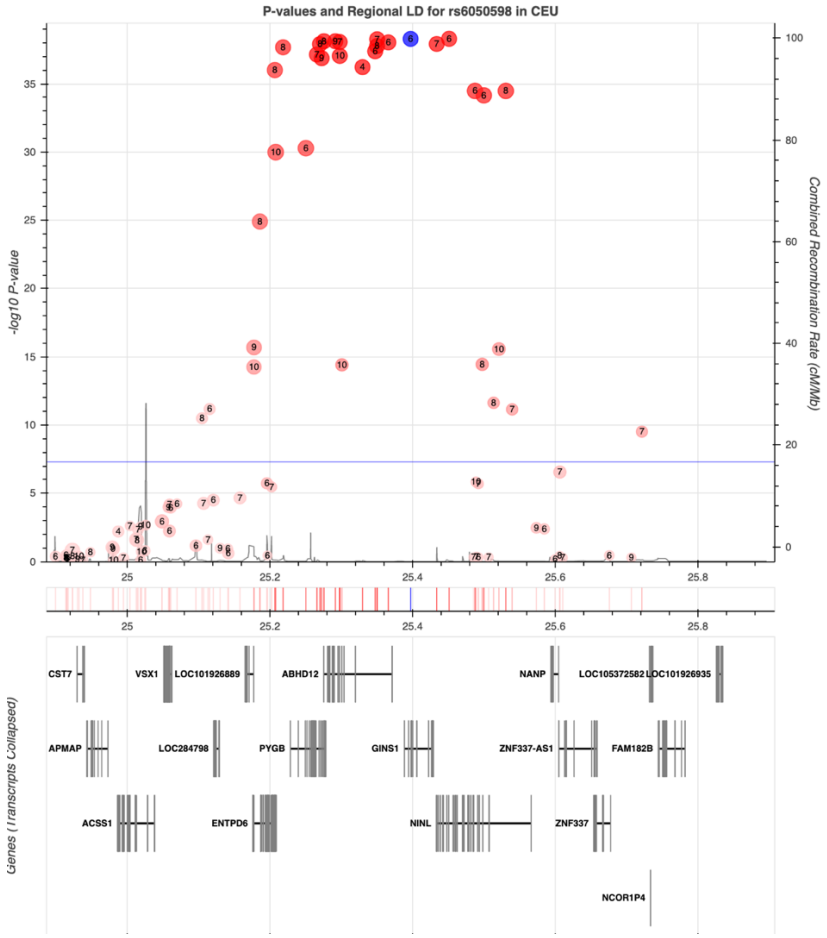
Upload your results using Browser (preferred: Chrome, Safari, Firefox 36+ or Internet Explorer). In real scenarios, you may want to upload only the information of a region of interest that you want to plot or only one chromosome because the upload time will be slow with very large files.

4. Select the columns that contain the information on Chromosome, Position, and P-Value.
5. Select the variant you want to center the plot on, and select the 1000 Genomes sub-population to plot recombination rates surrounding that variant.

LDassoc has three options for visualizing regions of association: by gene, by region or by variant. We will select visualising our results by using our lowest p-value variant as the index variant (rs449370). Select the CEU population (European, Utah Residents from North and West Europe) as the 1000 Genomes sub-population of interest. LDlink will calculate measures of linkage disequilibrium according to this population, which is the one that best matches the ancestry of our study population. Leave the rest of the options as default and press Calculate.



5. Explore the interactive plot.
Is the index variant in high LD with any of the nearby variants? Is it around areas of high or low recombination?
If LD were much stronger and widespread, how would it change this plot?
What if the strongest locus was not in LD with any other loci? How would that change your plot?
Remember the R2 is a measure of linkage disequilibrium or correlation of alleles for two genetic variants.



If you have gotten this far with time left, here are a couple of other challenges:

1. Go back to your logistic regression results and plot them:

1.5. Plot the results from 1.4. You will need to use the commands you ran in section two to generate the intermediate files for RStudio and the interactive browser plots. Your challenge will be to adapt those commands to the right files in the right sequence, run the RStudio script again, and create the plots.

2. What does including the PC covariates do? Try running it without any PC covariates.

Why would you want to include PCs in your analysis?