# GWAS practical

### Led by Alicia Martin (armartin@broadinstitute.org)

Today we will run several GWAS of simulated phenotypes using the 1000 Genomes data generated on the Affy6 genotyping array. A lot of 1000 Genomes data exists and openly accessible to anyone—this includes a lot of whole exome and genome sequence data, as well as genotyping array data on the Illumina Omni2.5 and Affy6.0 platforms.

The 1000 Genomes phase 3 (final) dataset consists of 2,504 individuals from 26 different populations across Africa, Europe, East Asia, South Asia, and the Americas, which you can learn more about here: <http://www.internationalgenome.org/category/population/>. The final release of sequencing data includes all unrelated individuals.

The genotype data has more individuals (including relatives), that are almost entirely overlapping, with 3450 individuals from these same populations. The dataset we will use today has 226,227 SNPs. For future reference, you could download all of the raw data with many more SNPs than we will use today from this website (but don’t worry about this right now since it’s already in your docker or on your laptop): <http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/supporting/hd_genotype_chip/>

We will start with two phenotypes, simulated height “measured” in centimeters across globally diverse populations, and normalized height with an average around 0 and standard deviation around 1. We have simulated taller Africans and Europeans, shorter Americans and East Asians, average South Asians, taller men, and shorter women. We also simulated a random diabetes case/control phenotype, and a glucose phenotype that is correlated with diabetes case status.

## Getting started

We first need to get back into the folder with the data. For docker, this should be here (but may be elsewhere if you’re not using docker):

cd /home/ginger/1kg

We also need to be able to use plink within the folder containing the GWAS data. To make it easier to run the plink program, we will add what’s called a *symbolic link* to the program. Depending on the path to plink, this should be run like this, but you may need to modify the path on your computer:

ln -s /home/ginger/plink/plink plink

A lot of iterative quality control goes into GWAS. To understand why this is necessary, we will first run a GWAS with no QC, and will later compare the results of this GWAS with a GWAS including a lot of SNP and sample QC.

Before running a GWAS, it’s helpful to know what our phenotype is. In plink, we can define a phenotype as the 6th column of a \*.fam or \*.ped file. Sometimes we may have multiple phenotypes, however, and it’s easier to create another file so that we don’t have to manually edit these files (which is not advised!) with the potential issue of a miscoded phenotype, or worse, missing/additional lines that corrupt the files. Plink uses phenotype files that contain 3 columns, one row per individual, as follows:

Family ID

Individual ID

Phenotype

We need to download a few of these files for the practical. To do this, download them like we have been. (Hint: remember that using the up key on your keyboard will go to the previous command, reducing how much you have to type)

wget broad.io/kemri/height\_raw.pheno

wget broad.io/kemri/height\_norm.pheno

wget broad.io/kemri/diabetes.pheno

wget broad.io/kemri/glucose.pheno

wget broad.io/kemri/CHS.inds

wget broad.io/kemri/phenotypes.txt

wget broad.io/kemri/ALL.wgs.nhgri\_coriell\_affy\_6.20140825.genotypes\_has\_ped.225k\_sites.fam

The first phenotype that we will use is in the height\_raw.pheno file. Take a look at this file as follows:

less height\_raw.pheno

This file should look something like this:

GBR HG00096 171.683923670094

GBR HG00097 163.617086718671

GBR HG00099 168.308802189014

GBR HG00100 166.64026671798

GBR HG00101 186.415964608269

Let’s try running this very simple GWAS with no QC. To do this, we will want to use the --assoc flag to run a very basic GWAS not including any covariates. The output of a plink call with this flag will generate an output file that ends in .assoc. This output file from a case/control analysis will contain the following information:

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)

We can run the GWAS as follows (*hint #1*: bash interprets a “\” character as instructions to move onto the next line, so lines ending with a backslash include information from the next lines. We use this only for readability. *Hint #2*: tab completion will be very helpful throughout this tutorial for some very long filenames!):

./plink \

--bfile ALL.wgs.nhgri\_coriell\_affy\_6.20140825.genotypes\_has\_ped.225k\_sites \

--assoc \

--pheno height\_raw.pheno \

--out ALL.height\_raw

We ran a GWAS! It’s kind of a pretty terrible GWAS with no SNP QC for missingness or allele frequency, and no sample QC considering relatedness or ancestry, however. Let’s take a look anyways.

less ALL.height\_raw.qassoc

The top 10 lines of the file should look something like this:

CHR SNP BP NMISS BETA SE R2 T P

1 rs10458597 564621 2475 -2.903 1.113 0.002743 -2.608 0.009154

1 1:564773:C:T 564773 2284 1.764 1.013 0.001327 1.741 0.0818

1 rs11240776 765269 2487 0.4709 1.396 4.579e-05 0.3373 0.7359

1 rs2980319 777122 2485 0.8542 0.2468 0.0048 3.461 0.0005481

1 rs2905036 792480 2487 1.723 0.7447 0.00215 2.314 0.02075

1 rs2341354 918573 2446 0.9157 0.2189 0.007108 4.183 **2.98e-05**

1 rs4970403 926431 2487 1.281 0.3553 0.005206 3.606 0.000317

1 rs2465136 990417 2468 0.9576 0.2274 0.007139 4.211 **2.634e-05**

1 rs2710872 990517 2369 0.8774 0.4469 0.001626 1.963 0.04974

1 rs4075116 1003629 2463 0.5412 0.2465 0.001955 2.196 0.02822

Hmm, notice that we \*already\* have some really significant p-values. By chance, with α=0.05, we expect our multiple test correction threshold to be 0.05 / 10 tests = 0.005. This indicates that our test is likely poorly calibrated, which we will come back to later. For now, let’s start running some variant QC.

## Allele frequency filtering

GWAS are not well-powered to test associations with rare variants. We will start by filtering out sites not meeting some allele frequency threshold. Plink stores alleles in major/minor format. Consistent with some large, previously published GWAS studies, such as the Psychiatric Genomics Consortium Schizophrenia Working Group paper published in Nature in 2014, we will start by filtering out variants with a minor allele frequency less than 2%, as follows:

./plink --bfile ALL.wgs.nhgri\_coriell\_affy\_6.20140825.genotypes\_has\_ped.225k\_sites \

--geno 0.02 \

--make-bed \

--out ALL.geno02

*Note:* we have changed the output filename so that we don’t overwrite the original file, which we should use for downstream intermediate steps.

*Note 2*: this was a minor mistake. The --maf flag filters based on minor allele frequency, whereas the --geno flag actually filters for genotype missingness. Both are useful in GWAS.

Bonus: we may also in the future want to filter out individuals with high missingness. This is unlikely to be an issue in the 1000 Genomes data, but is useful to check for GWAS generally.

**Question 1)** According to the log file, how many SNPs did we remove? Hint: this is either on your screen after running this command, or can be found in the ALL.geno02.log file.

**Question 2)** What fraction of sites were removed from the original file (i.e. the ALL.wgs.nhgri\_coriell\_affy\_6.20140825.genotypes\_has\_ped.225k\_sites plink file)?

## Check the reported versus genetic sex

We are often interested in making sure that manually curated information lines up with genetically inferred information in case there are inevitable human entry errors. One example of this type of information is sex. We can test heterozygosity on the X chromosome to genetically infer sex using the --check-sex flag, and see how this lines up using plink’s self-reported pedigree information. This will produce a file with the following information:

FID Family ID

IID Individual ID

PEDSEX Sex as determined in pedigree file (1=male, 2=female)

SNPSEX Sex as determined by X chromosome

STATUS Displays "PROBLEM" or "OK" for each individual

F The actual X chromosome inbreeding (homozygosity) estimate

A “PROBLEM” arises if the two sexes do not match, or if the SNP data or pedigree data are ambiguous with regard to sex. A male call is made if F is more than 0.8; a female call is made if F is less than 0.2. Try running this with our data:

./plink --bfile ALL.geno02 \

--check-sex \

--out ALL.geno02.sex

**Question 3)** How many PROBLEM samples do we have?

*Hint*: you can use grep and wc with the output file to answer this question.

We have a lot of mismatches. Part of the reason is that we have a very heterogeneous dataset. Most GWAS are done using a single population, but our data has 26 globally diverse populations. This makes heterozygosity calculations hard to interpret. By running the following commands, we can see that plink is not actually making a call for the vast majority of the PROBLEM individuals:

less ALL.geno02.sex.sexcheck

grep PROBLEM ALL.geno02.sex.sexcheck | sort -k4r | head

Instead, only the first few individuals actually look problematic:

CEU NA12865 2 1 PROBLEM 0.9598

ESN HG03511 2 1 PROBLEM 0.9396

LWK NA19332 2 1 PROBLEM 0.9316

TSI NA20530 2 1 PROBLEM 0.9172

CEU NA10854 2 1 PROBLEM 0.9049

FIN HG00361 2 1 PROBLEM 0.8707

**YRI NA19176 2 0 PROBLEM 0.7936**

**TSI NA20533 2 0 PROBLEM 0.782**

**YRI NA19226 1 0 PROBLEM 0.7684**

**CEU NA07348 2 0 PROBLEM 0.7594**

(Note the 0’s in the 4th SNPSEX column, where plink has not made a call for an individual’s sex.)

In an actual GWAS study, we wouldn’t have such heterogeneous ancestries in one GWAS. Instead, we would usually define homogeneous clusters of ancestry and perform QC within these ancestry clusters. For today’s purposes, we will confirm this by looking at a single population that was inferred to have the most PROBLEM individuals when inferred with all global 1000 Genomes populations.

First, to see how many individuals there are (one per line) whose genetic sex does not match their reported sex in the CHS population, we will run the following:

grep PROBLEM ALL.geno02.sex.sexcheck | grep CHS | wc -l

**Question 4)** How many CHS individuals have genetic sex that does not match their reported sex when we consider this with all other populations?

**Question 5**) How many total CHS individuals did we start with?

*Hint*: use a similar series of commands, including grep and wc, as above.

To see how this behaves in a single population, we will *keep* only the CHS individuals from this larger dataset. Note that --keep is for individuals and --extract is for SNPs. --keep requires a list of Family ID / Individual ID pairs, one set per line, i.e. one person per line

./plink --bfile ALL.geno02 \

--keep CHS.inds \

--make-bed \

--out CHS.geno02

Now, we will run the sex check on these individuals from the CHS population only. Try to run the sex check on these individuals as follows:

./plink --bfile CHS.geno02 \

--check-sex \

--out CHS.geno02.sex

**Question 6**) How many individuals have a reported sex that differs from their genetically inferred sex now that we are only looking at a single population (the CHS)?

*Hint*: recall how we checked this in questions 4 and 5.

## Filter SNPs in LD

Some QC metrics that we will need to compute, such as relatedness or ancestry (via principal components), assume independent SNPs are measured. This means that they are not genetically correlated or in linkage disequilibrium (LD). We measure LD for pairs of nearby SNPs using the --indep-pairwise flag to compute pairwise genotypic correlation. This flag requires 3 numeric arguments, as follows:

1. window size in SNPs (e.g. 50)
2. the number of SNPs to shift the window at each step (e.g. 5)
3. the r2 threshold

Using this flag will generate the following files:

\*.prune.in

\*.prune.out

Because this computes pairwise statistics on many SNPs, this takes a few seconds. Because of this, filtering out SNPs in LD is a two-step process. The first step is determining a list of SNPs in LD, and the second step is actually extracting independent SNPs. Let’s try filtering out SNPs in LD from the dataset we have QC’d up until now with all populations:

./plink --bfile ALL.geno02 \

--indep-pairwise 50 5 0.2 \

--out ALL.geno02.ld

This generates an LD file with the filename from --out and the endings described above.

**Question 7**) How many SNPs will be removed due to LD filtering?

Now, we need to remove those SNPs from our QC file, as follows:

./plink --bfile ALL.geno02 \

--extract ALL.geno02.ld.prune.in \

--make-bed \

--out ALL.geno02.ld20

## Infer relatedness and filter to unrelated individuals

Now that we have independent SNPs, we would like to compute pairwise relatedness among all pairs of individuals. We can do this using the --genome flag. It produces a lot of output, as follows:

FID1 Family ID for first individual

IID1 Individual ID for first individual

FID2 Family ID for second individual

IID2 Individual ID for second individual

RT Relationship type given PED file

EZ Expected IBD sharing given PED file

Z0 P(IBD=0)

Z1 P(IBD=1)

Z2 P(IBD=2)

PI\_HAT P(IBD=2)+0.5\*P(IBD=1) ( proportion IBD )

PHE Pairwise phenotypic code (1,0,-1 = AA, AU and UU pairs)

DST IBS distance (IBS2 + 0.5\*IBS1) / ( N SNP pairs )

PPC IBS binomial test

RATIO Of HETHET : IBS 0 SNPs (expected value is 2)

The Z0, Z1, and Z2 columns are the probabilities that the pair of individuals share 0, 1, or 2 chromosomes, respectively. These probabilities are independent and must therefore sum to 1. PI\_HAT is the most important column for filtering to unrelated individuals, and it simply gives the proportion of identity-by-descent (i.e. very rough approximation of the fraction of shared DNA). We filter out one individual from pairs who share PI\_HAT > 0.1.

Unfortunately for relatedness, plink can be a little tricky, slow, and/or not the best software in some scenarios due to population structure. We have a ton of population structure in our dataset, since we are looking at 26 globally diverse populations. A better tool that works well with plink files is called king. You can learn more about king here: <http://people.virginia.edu/~wc9c/KING/manual.html>.

You can try running this as follows (if this installation doesn’t work right now, we have the output stored for you, and you can always come back and spend more time on this step).

wget http://people.virginia.edu/~wc9c/KING/Linux-king.tar.gz

tar zxvf Linux-king.tar.gz

./king -b ALL.geno02.ld20.bed \

--unrelated

If you run into problems, try to just grab these files, as follows:

wget broad.io/kemri/kingunrelated.txt

wget broad.io/kemri/kingunrelated\_toberemoved.txt

First, look at the files with less. Next, let’s check how many individuals we removed from each population.

cut -f1 kingunrelated\_toberemoved.txt | sort | uniq -c

28 ACB

56 ASW

45 BEB

9 CDX

68 CEU

68 CHS

45 CLM

66 ESN

8 GBR

8 GIH

63 GWD

55 IBS

8 ITU

23 KHV

25 LWK

38 MSL

40 MXL

41 PEL

56 PJL

43 PUR

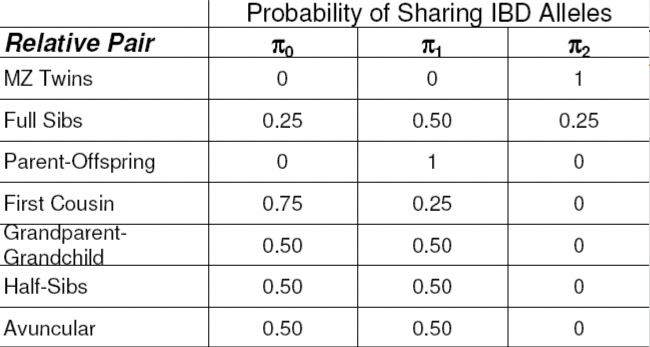
20 STU

4 TSI

63 YRI

**Question 8**) Using similar commands, how many total individuals did we remove, without considering population?

Now we can remove individuals who are 2nd degree relatives or closer. Recall relatedness from before:



Remove one individual per pair from close relatives, again using the --keep flag, as follows:

./plink --bfile ALL.geno02 \

--keep kingunrelated.txt \

--make-bed \

--out ALL.geno02.unrel

We need to also assess pairwise relatedness among these individuals also for use in PCA, again using the --genome flag, as follows:

./plink --bfile ALL.geno02.unrel \

--genome \

--out ALL.geno02.unrel

## Compute PCA to correct for population stratification

Now we would like to look at ancestry in unrelated individuals. The most commonly used method to correct for ancestry stratification is called principal components analysis (PCA). Each component from smallest to largest explains increasing amounts of genetic variation. In other words, PC1 explains more variation than PC2, which explains more variation than PC3, and so on. Multi-dimensional scaling can be a mathematically equivalent way of computing PCs. Let’s use plink to run MDS, as follows:

./plink --bfile ALL.geno02.unrel \

--read-genome ALL.geno02.unrel.genome \

--cluster \

--K 2 \

--mds-plot 10 \

--out ALL.geno02.unrel.mds

The --K flag request that the clustering process stops at a certain fixed number of clusters. The --mds-plot flag specifies how many principal components to compute. We have computed 10 PCs in the above command. --mds-plot produces the following output with one row per individual:

FID Family ID

IID Individual ID

SOL Assigned solution code (from --cluster)

C1 Position on first dimension

C2 Position on second dimension

C3 Position on third dimension

C4 Position on fourth dimension

…

As an aside, if we also want to know where relatives would be in PCA space, we could use the PCA loadings to "project" related individuals. For now, we have excluded all related individuals from our GWAS analysis.

## Hardy-Weinberg equilibrium

Normally, we would want to remove SNPs that are out of Hardy-Weinberg equilibrium, because they are more likely to be enriched for technical errors. However, the calculation of this test involves heterozygosity, which is problematic in a dataset containing globally diverse populations, as we discussed in during the sex check section. Why?

The assumptions of Hardy-Weinberg equilibrium are:

✓ organisms are diploid✓ only sexual reproduction occursX generations are non-overlapping

X mating is randomX population size is infinitely large

X allele frequencies are equal in the sexesX there is no migration, mutation, or selection

These assumptions are clearly violated in humans, especially in datasets with multiple diverse populations. If you consider a dataset for GWAS in the future, it will most likely be more ancestrally homogeneous. You would use the --hardy flag to remove potentially problematic SNPs, which would generate a file containing the following information:

SNP SNP identifier

TEST Code indicating sample

A1 Minor allele code

A2 Major allele code

GENO Genotype counts: 11/12/22

O(HET) Observed heterozygosity

E(HET) Expected heterozygosity

P H-W p-value

You might then remove those SNPs that are clearly outside HWE, often determined in GWAS by p < 1e-5 from this type of test.

## Filter out bad phenotypes

We have talked earlier about having filtering out phenotypes that are big outliers. When we looked at height earlier, we saw that there were likely some manual curation individuals, as most individuals are probably not ~50 cM tall.

*Try on your own after class*: How would you filter out individuals with bad phenotypes? You might use R to visualize the phenotypes, then filter to individuals who have a reasonable reported height. You could write a file that has these family IDs and individual IDs, then use plink’s --remove flag (opposite of the --keep flag) to remove individuals with outlier phenotypes.

## Run QC’d GWAS!

We’re skipping the outlier phenotype filtering step as an exercise to do at home, but done the vast majority of the QC—we have filtered SNPs and smaples, and generated the covariates that we will need to correct for some population stratification. We’ve compiled a covariates file for you using output that you generate in this tutorial, which you will need to download as follows:

wget broad.io/kemri/ALL.covariates

Because we will now include covariates with a continuous phenotype, we will now use the --linear flag instead of the --assoc flag. The --linear output contains the following:

CHR Chromosome

SNP SNP identifier

BP Physical position (base-pair)

A1 Tested allele (minor allele by default)

TEST Code for the test (see below)

NMISS Number of non-missing individuals included in analysis

BETA/OR Regression coefficient (--linear) or odds ratio (--logistic)

STAT Coefficient t-statistic

P Asymptotic p-value for t-statistic

Let's rerun this association analysis on the unrelated, QC'd samples with sex and 10 PCs as covariates

./plink --bfile ALL.geno02.unrel.maf01 \

--pheno height\_raw.pheno \

--covar ALL.covariates \

--hide-covar \

--linear \

--out ALL.geno02.unrel.maf01.height\_raw

Voila! We will compare the QC’d and confounded analysis later in R.