GWAS Assignment

Conor O'Donoghue

December 18, 2020

1. Overview

To start with, it would be best to get a summary of the cohort we're going to be studying.

The new version of plink requires arguments beyond an input, but with version 1.07 you can just provide the input files and it will give you summary stats for them. So just for this first step, I'll load the old version of plink.

```
module load plink/1.07
plink --bfile gwas --noweb
```

The option –noweb is required because the old version of plink automatically tries to connect to the web and update itself, which we don't want.

The output is as follows:

```
Reading map (extended format) from [ gwas.bim ]
306102 markers to be included from [ gwas.bim ]
Reading pedigree information from [ gwas.fam ]
4000 individuals read from [ gwas.fam ]
4000 individuals with nonmissing phenotypes
Assuming a disease phenotype (1=unaff, 2=aff, 0=miss)
Missing phenotype value is also -9
2000 cases, 2000 controls and 0 missing
2000 males, 2000 females, and 0 of unspecified sex
Reading genotype bitfile from [ gwas.bed ]
Detected that binary PED file is v1.00 SNP-major mode
Before frequency and genotyping pruning, there are 306102 SNPs
4000 founders and 0 non-founders found
Total genotyping rate in remaining individuals is 0.983323
0 SNPs failed missingness test ( GENO > 1 )
0 SNPs failed frequency test ( MAF < 0 )
After frequency and genotyping pruning, there are 306102 SNPs
After filtering, 2000 cases, 2000 controls and 0 missing
After filtering, 2000 males, 2000 females, and 0 of unspecified sex
```

It seems that there are 2000 each of cases, controls, males, and females. All 4000 individuals are founders (meaning that they are not indicated as related), which means that there are no families. In total, there are 306102 SNPs.

2. QC tests

There are a number of QC tests we can run to check the quality of the data. Afterwards, we may choose to filter out data that do not pass certain thresholds.

Data Missingness

We can get summaries of the missingness of the data using the following command (after a quick swap to the most up-to-date plink version)

```
module unload plink
module load plink
plink --bfile gwas --missing --out gwas_miss
```

This will create two output files: gwas_miss.imiss, and gwas_miss.lmiss.

Individual missingness

gwas_miss.imiss contains the missingness of the data at the **i**ndividual level. The first few lines look like this:

```
FID     IID MISS_PHENO   N_MISS   N_GENO   F_MISS
   0   A2001          N     5168   306102  0.01688
   1   A2002          N     5175   306102  0.01691
   2   A2003          N     5150   306102  0.01682
   3   A2004          N     5076   306102  0.01658
```

FID is the family ID, and IID is the individual ID. Since each individual in the sample is a founder, FID and IID are not the same, but they should be redundant.

MISS_PHENO is whether or not (Y/N) the individual is missing the phenotype.

N_MISS is the number of SNPs for which that individual is missing data, and F_MISS is the proportion of SNPs that are missing data for that individual.

N_GENO is the number of non-obligatory missing genotypes. Some samples may have only been genotyped on a subset of SNPs, so there may be cases where data is missing on purpose. This is essentially the denominator for the calculation in F_MISS, so that we use total SNPs *for that individual* rather than total SNPs in the dataset.

2i) How many SNPs is individual A2038 missing data for?

We can figure this out by simply using grep to look for the line containing A2038:

```
cat gwas_miss.imiss | grep A2038
```

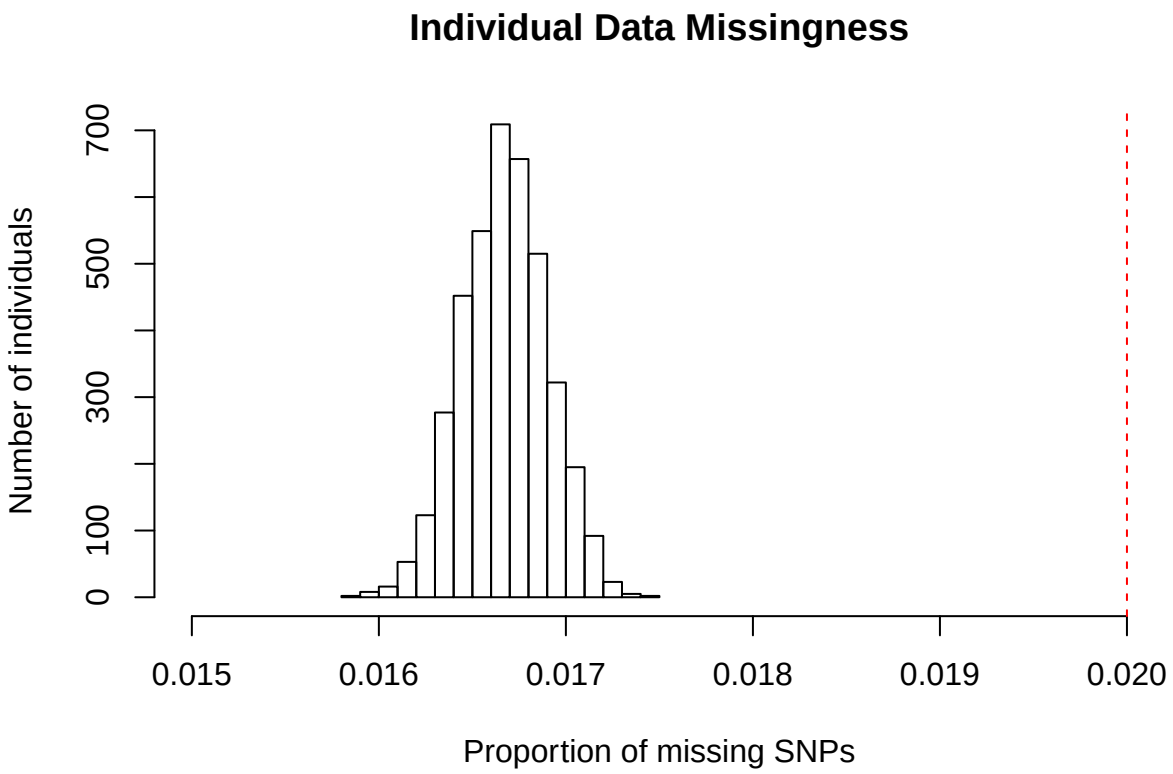
Which returns the following:

```
37   A2038          N     5211   306102  0.01702
```

It seems this individual is missing 5211 SNPs, which is 1.7% of all the SNPs in our dataset.

We can make a histogram of the F_MISS column, which will give us a better understanding of the overall missingness of the data at the individual level.

```
imiss <- read.csv('gwas_miss.imiss', header=T, sep='')
hist(imiss$F_MISS, main="Individual Data Missingness",
     ylab="Number of individuals",
     xlab="Proportion of missing SNPs",
     xlim=c(0.015,0.020))
abline(v=0.02, lty=2, col='red')
```



Just about all of the individuals are missing between 1.6% and 1.75% of all SNPs. As the default value for missingness that would result in filtering is 0.1, and the QC paper we were given suggests that using a threshold of 0.02 would be a strict test, this appears to be a low missingness rate.

SNP missingness

The other file output with the –missing command in plink is gwas_miss.lmiss, which contains missingness of the data at the SNP (**l**ocus) level.

The first few lines look as follows:

```
 CHR         SNP   N_MISS   N_GENO   F_MISS
   1   rs3934834      182     4000   0.0455
   1   rs3737728       18     4000   0.0045
   1   rs6687776       85     4000  0.02125
   1   rs9651273       57     4000  0.01425
```

We now have CHR for chromosome ID and SNP for SNP ID, but N_MISS, N_GENO, and F_MISS are the same as before – now just from the perspective of each SNP rather than each individual.

2ii) For how many individuals is the SNP rs2493272 data missing?

We can look for rs2493272 in the lmiss file as follows:

```
cat gwas_miss.lmiss | grep rs2493272
```

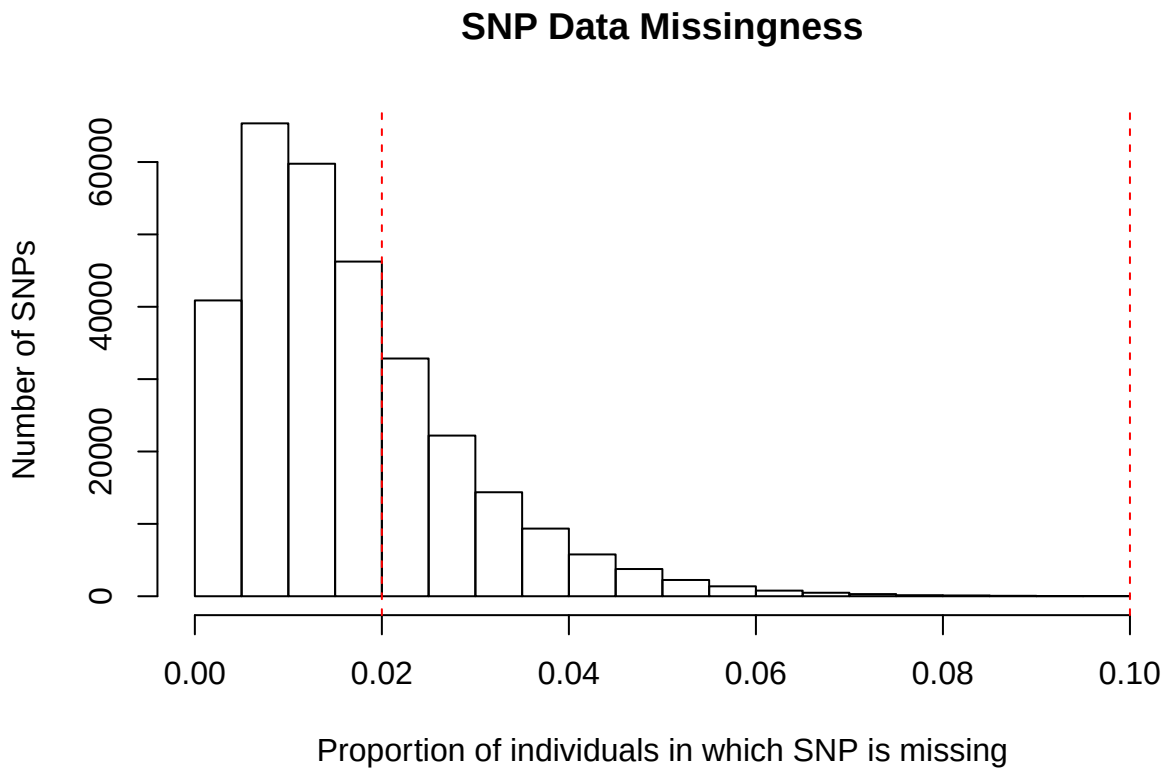
Which returns:

```
   1   rs2493272      111     4000  0.02775
```

It seems that this SNP is missing in 111 individuals, resulting in a missingness rate of 2.775%.

We can make a similar plot as before with the lmiss F_MISS column, which will give us a better view of missingness rates at the SNP level.

```
lmiss <- read.csv('gwas_miss.lmiss', header=T, sep='')
hist(lmiss$F_MISS, main="SNP Data Missingness",
     ylab="Number of SNPs",
     xlab="Proportion of individuals in which SNP is missing")
abline(v=0.02, lty=2, col='red')
abline(v=0.1, lty=2, col='red')
```



There's a much wider range of missingness at the SNP level than there is at the individual level. If we went with the default values and filtered out SNPs that are missing in at least 10% of individuals, hardly any SNPs would be removed. But if we were to go with the stricter threshold of 2%, then we would cut out a sizeable amount of the data.

3. Allele frequencies

We can obtain the frequencies of each allele with the following command:

```
plink --bfile gwas --freq --out gwas_freq
```

The resulting file, gwas_freq.frq, contains the following data:

```
 CHR         SNP   A1   A2          MAF  NCHROBS
   1   rs3934834    T    C      0.09966     7636
   1   rs3737728    A    G       0.3386     7964
   1   rs6687776    T    C      0.06117     7830
   1   rs9651273    A    G       0.4065     7886
```

CHR and SNP contain the Chromosome and SNP IDs as before, but now we have A1 and A2, which are the base code of the minor and major alleles respectively.

MAF is the frequency of the minor allele (A1), and NCHROBS is the count of non-missing alleles (if no alleles were missing there would be 8000 – two for each individual).

3i) Which is the minor allele for SNP rs4970357 and what is its frequency? We can find this SNP with grep as before:

```
cat gwas_freq.frq | grep rs4970357
```

Which gives the following row:

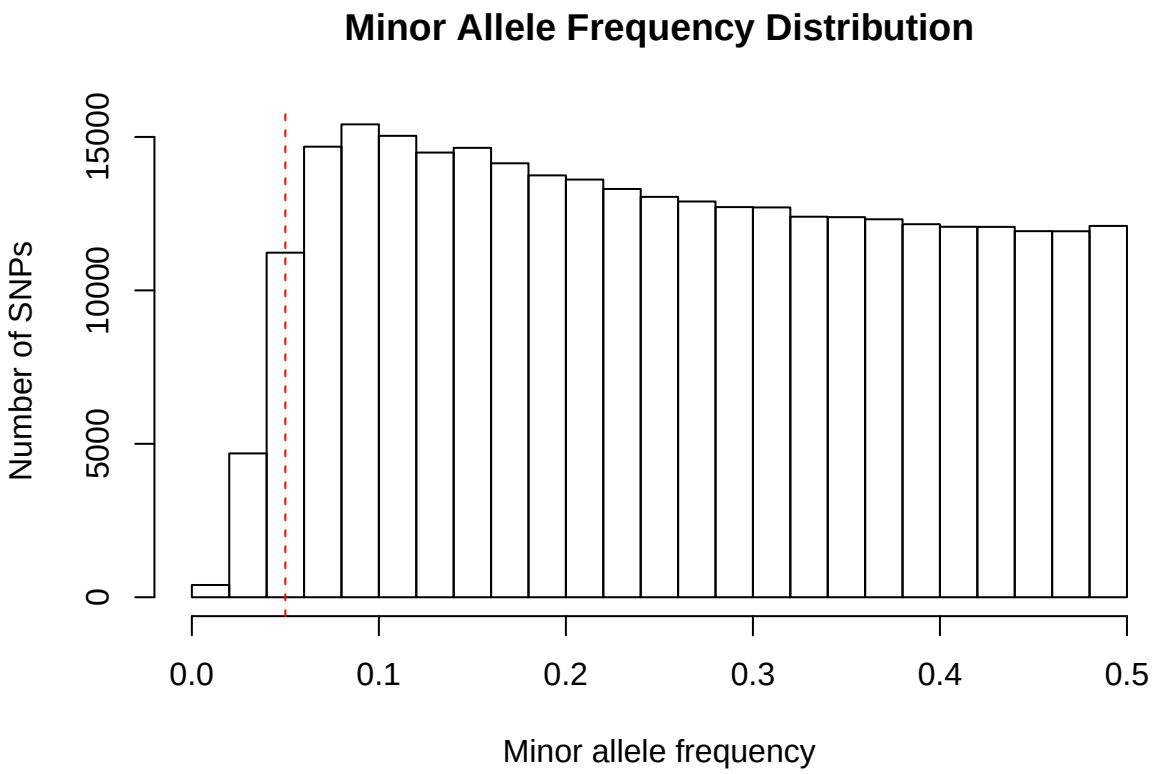
```
1   rs4970357    C    A      0.05028     7856
```

This SNP's minor allele is a C, and its frequency is about 0.05.

3ii) Create a plot which shows the overall distribution of MAF.

Similarly to the missingness plots, we can plot a histogram of the MAF column to get an idea of how rare each minor allele is.

```
frq <- read.csv('gwas_freq.frq', header=T, sep='')
hist(frq$MAF, breaks=25, main="Minor Allele Frequency Distribution",
     xlab="Minor allele frequency", ylab="Number of SNPs")
abline(v=0.05, lty=2, col='red')
```



There are almost no rare alleles (frequency < 0.05), which seems strange, as in lecture as well as the practical there was a spike of alleles near 0.0, which was the biggest bar in the histogram. This means there isn't much to remove, but it's quite strange. Between this and the missingness tests, this dataset started out very clean. Is it possible that some amount of QC was done on the data already before it was given to us?

4. Other QC steps

The assignment requires us to carry out two additional QC steps, and visualize any results.

Sex discrepancy

I had started by trying to investigate sex discrepancy with the following command:

```
plink --bfile gwas --check-sex --out gwas_sex
```

But received the following error:

```
Error: --check-sex/--impute-sex requires at least one polymorphic X chromosome locus.
```

The goal would have been to examine heterozygosity of alleles on the X chromosome to estimate whether the individual is male or female, and see if there is any discrepancy between what sex they are identified as in the individual data and what sex they appear to be through their X chromosome alleles.

However, it seems that this dataset doesn't contain any alleles on the X chromosome, so the –check-sex command doesn't work. It could be that I'm using the command incorrectly, but both the QC paper and

the plink documentation recommended the command above, so I'll take it at its word and try two different QC commands.

Hardy-Weinberg equilibrium

SNPs which deviate from hardy-weinberg equilibrium are a common indicator of genotyping error, so it's a good idea to look at HWE p-values of each SNP.

We can obtain HWE data using the following command:

```
plink --bfile gwas --hardy --out gwas_hardy
```

Which creates the file gwas_hardy.hwe. The first few lines of the resulting file are the following:

```
CHR         SNP     TEST   A1   A2                 GENO   O(HET)   E(HET)            P
   1   rs3934834      ALL    T    C          46/669/3103   0.1752   0.1795       0.1486
   1   rs3934834      AFF    T    C           23/348/1582   0.1782   0.1814       0.4528
   1   rs3934834    UNAFF    T    C           23/321/1521   0.1721   0.1774       0.1919
   1   rs3737728      ALL    A    G        428/1841/1713   0.4623   0.4479      0.04379
```

As before CHR and SNP contain chromosome and SNP IDs, and A1 and A2 contain the base code for the minor and major alleles respectively.

For each SNP, three tests are ran: ALL (all individuals), AFF (cases only), and UNAFF (controls only). Which test was run is contained in the TEST column.

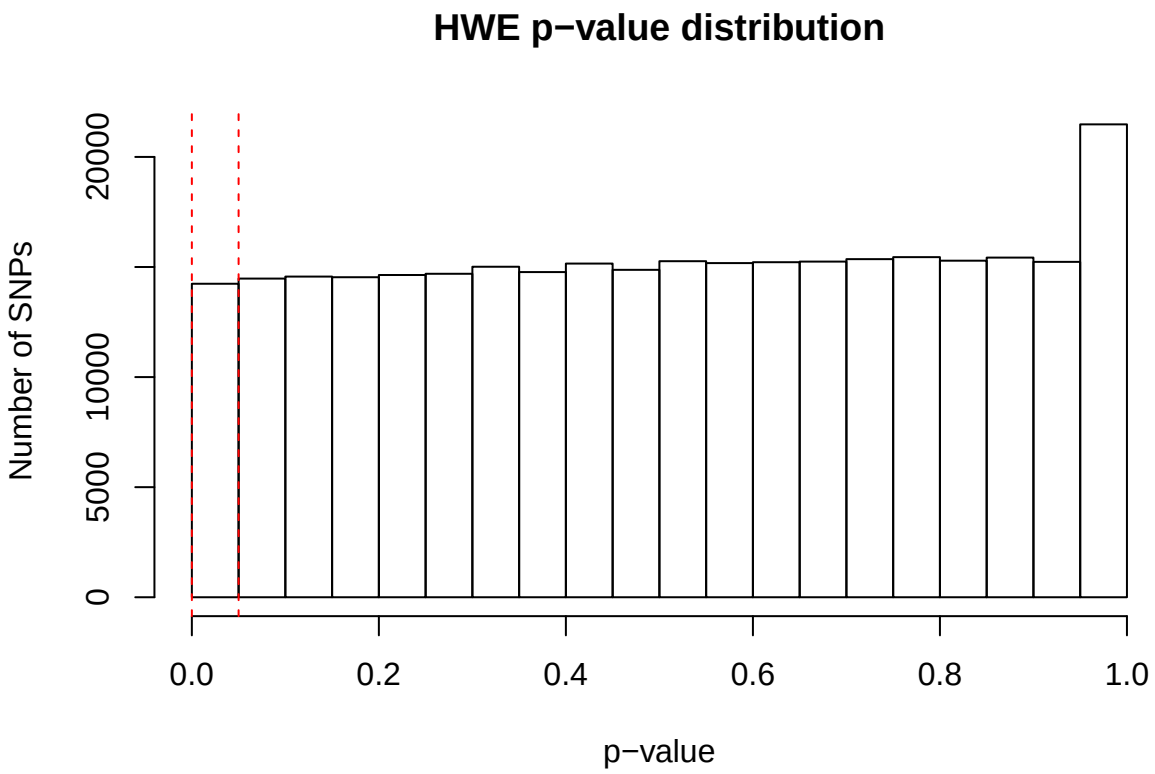
GENO contains the counts for each genotype: 11/12/22. So in the first row, there were 46 individuals of genotype TT, 669 of TC, and 3103 of CC.

O(HET) contains the observed heterozygosity, and E(HET) contains the expected heterozygosity. The final column, P, contains the HW p-value.

We're interested in the p-values of the ALL tests (since it appears that that's the value PLINK checks against the threshold for filtering), which we can plot as follows:

```
hwe <- read.csv('gwas_hardy.hwe', header=T, sep='')
```

```
hwe_all <- hwe[which(hwe$TEST == 'ALL'),]
hist(hwe_all$P, breaks=25, main="HWE p-value distribution",
     xlab="p-value", ylab="Number of SNPs")
abline(v=0.05, lty=2, col='red')
abline(v=0.000001, lty=2, col='red')
```



Though the HWE is rejected at p<0.05, we have to keep in mind the scale of the testing. The QC paper we were given actually suggests to exclude SNPs if they have a HWE p-value of less than 1e-6. From the graph that appears to be a very small proportion of the total SNPs, so let's look at the SNPs with the lowest p-values and see how many SNPs we would exclude under that criteria:

```
head(hwe_all[order(hwe_all$P),])
```

```
##        CHR        SNP TEST A1 A2          GENO O.HET. E.HET.         P
## 7828     1   rs925826  ALL  A  G   59/590/3222 0.1524 0.1662 2.389e-06
## 676120  13  rs7987203  ALL  T  C 900/1830/1239 0.4611 0.4964 7.537e-06
## 256579   4  rs2315621  ALL  G  A 377/1886/1734 0.4719 0.4424 2.432e-05
## 488950   8  rs7011064  ALL  C  T   23/812/3053 0.2088 0.1963 2.668e-05
## 868030  20  rs1887320  ALL  G  A 478/1936/1465 0.4991 0.4676 2.792e-05
## 572152  10 rs11191092  ALL  C  T 146/1443/2358 0.3656 0.3430 2.848e-05
```

It seems only two SNPs have a HWE p-value of less than 1e-6, so once again we find ourselves needing to get rid of only a tiny amount of data.

Relatedness

The plink command –genome will do a pairwise calculation of identity by descent for each pair of individuals in the study. We can combine this with the –min command to set a threshold for relatedness, and creating a list of all individuals that have relatedness above the threshold.

For example, –min 0.2 sets the threshold at second-degree relatives, and creates a list of pairs of individuals that are at least as related as second-degree relatives.

I tried doing exactly that command:

```
plink --bfile gwas --genome --min 0.2 --out gwas_min
```

Which created an empty gwas_min.genome file – indicating that there were no pairs of individuals that were at least as related as second-degree relatives.

Though it's rather large, I also have a genome file containing all of the pairwise comparisons, which looks like the following:

```
 FID1   IID1 FID2   IID2 RT    EZ      Z0      Z1      Z2  PI_HAT PHE       DST     PPC   RATIO
   0  A2001   1  A2002 UN    NA  1.0000  0.0000  0.0000  0.0000   1  0.720609  0.5916  2.0147
   0  A2001   2  A2003 UN    NA  0.9769  0.0231  0.0000  0.0115   1  0.721555  0.9945  2.1698
   0  A2001   3  A2004 UN    NA  0.9720  0.0256  0.0024  0.0152   1  0.725143  0.7260  2.0385
   0  A2001   4  A2005 UN    NA  0.9959  0.0000  0.0041  0.0041   1  0.722814  0.5999  2.0161
```

As before there is a FID and IID for family and individual IDs, but now there's one for each individual, since we're doing a pairwise comparison.

RT contains the relationship given in the input PED file, which should be UN (unrelated) for every individual in our study. EZ is the expected IBD sharing, which is NA for unrelated individuals.

Z0, Z1, and Z2 are the probability of IBD being equal to 0, 1, or 2 respectively.

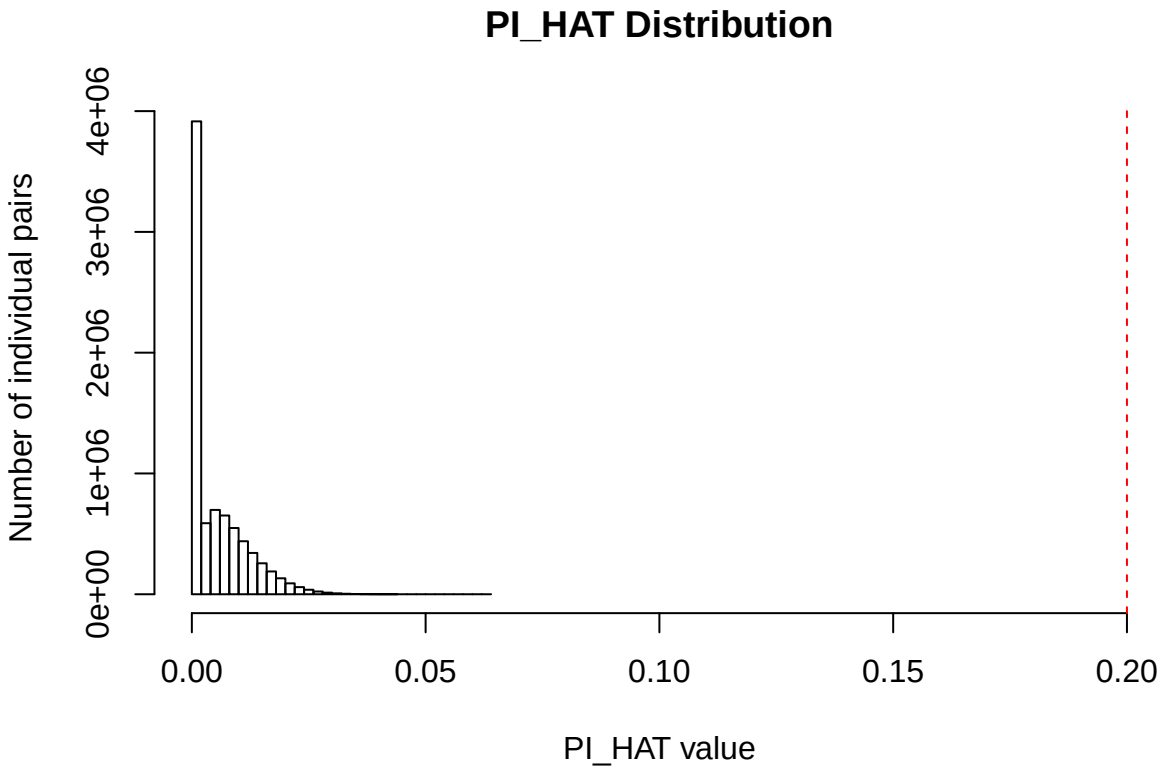
PI_HAT – what we actually want to investigate – is equal to Z2+0.5*Z1, aka the proportion IBD. This is the value that needed to be above 0.2 in the –min example above for the pair to be considered related.

Though not used in our threshold, PHE contains the pairwise phenotypic code, DST contains the IBS distance, PPC contains the IBS binomial test, and RATIO contains the ratio of HETHET : IBS 0 SNPs (with an expected value of 2).

We can plot PI_HAT to see if any pairs got anywhere close to the threshold of 0.2:

```
rel <- read.csv('gwas_genome.genome', header=T, sep='')
```

```
hist(rel$PI_HAT, breaks=25, main="PI_HAT Distribution",
     xlab="PI_HAT value", ylab="Number of individual pairs",
     xlim=c(0,0.2))
abline(v=0.2, lty=2, col='red')
```



It appears that none of the PI_HAT values get anywhere close to our relatedness threshold, so all of the individuals in our study should be well and truly unrelated.

5. Association Testing

Next we can finally run our association tests. One would normally filter the data to get rid of SNPs or individuals outside of our thresholds defined in the QC testing section with a command like the following:

```
plink --bfile gwas --maf 0.05 --mind 0.2 --geno 0.05 --hwe 0.000001 --out gwas_clean
```

but looking at our GWAS tests our starting data is very clean. I have a hunch that it's been cleaned already, but regardless it's definitely good to go as is.

Different Genetic Models

We can start using the –model command to test each SNP with multiple different models:

```
plink --bfile gwas --cell 0 --model --out gwas_mod
```

Which results in the file gwas_mod.model:

CHR	SNP	A1	A2	TEST	AFF	UNAFF	CHISQ	DF	P
1	rs3934834	T	C	GENO	23/348/1582	23/321/1521	0.2607	2	0.8778
1	rs3934834	T	C	TREND	394/3512	367/3363	0.1277	1	0.7209
1	rs3934834	T	C	ALLELIC	394/3512	367/3363	0.1307	1	0.7177
1	rs3934834	T	C	DOM	371/1582	344/1521	0.1906	1	0.6625
1	rs3934834	T	C	REC	23/1930	23/1842	0.02475	1	0.875
1	rs3737728	A	G	GENO	206/950/842	222/891/871	2.931	2	0.231
1	rs3737728	A	G	TREND	1362/2634	1335/2633	0.1778	1	0.6733
1	rs3737728	A	G	ALLELIC	1362/2634	1335/2633	0.172	1	0.6783
1	rs3737728	A	G	DOM	1156/842	1113/871	1.257	1	0.2623

As before, CHR and SNP contain the chromosome and SNP IDs, and A1 and A2 are the minor and major alleles respectively.

TEST contains the type of test, which can be either GENO (genotypic test, 2df), TREND (the Cochran-Armitage trend test), ALLELIC (the basic allelic test), DOM (dominant gene action test, 1df), or REC (recessive gene action test, 1df).

AFF and UNAFF contain the affected and unaffected genotypes (for the genotype test) or alleles (for the other tests) respectively. CHISQ contains the chi-squared statistic for the test, DF contains the degrees of freedom (Which should be 2 for the genotype test and 1 for the others), and P contains the p-value associated with the chisq statistic.

5i) Under which genetic model does SNP rs9651273 show the smallest p-value?

```
cat gwas_mod.model | grep rs9651273
```

1	rs9651273	A	G	GENO	322/1013/650	305/939/714	6.085	2	0.04773
1	rs9651273	A	G	TREND	1657/2313	1549/2367	3.995	1	0.04563
1	rs9651273	A	G	ALLELIC	1657/2313	1549/2367	3.892	1	0.04853
1	rs9651273	A	G	DOM	1335/650	1244/714	6.029	1	0.01407
1	rs9651273	A	G	REC	322/1663	305/1653	0.3062	1	0.58

The dominant gene action test resulted in the smallest p-value for SNP rs9651273, which means of the five models the one assuming dominance is the one in which the observed values were most unlikely to happen purely by chance.

The p-value is under 0.05, but we can't conclude that it's significant because we haven't corrected for multiple testing.

6. Correction for Multiple Testing

We can run an association test that corrects for multiple testing with the following command:

```
plink --bfile gwas --assoc --adjust --out as
```

This will create two different files: gwas_as.assoc, which is the same as just running plain –assoc without –adjust, and doesn't correct for multiple testing; and gwas_as.assoc.adjusted, which provides associations that have been corrected for multiple testing. Let's jump right in to the latter:

CHR	SNP	UNADJ	GC	BONF	HOLM	SIDAK_SS	SIDAK_SD	FDR_BH	FDR_B
3	rs6802898	2.327e-20	3.599e-20	7.123e-15	7.123e-15	7.123e-15	7.123e-15	7.123e-15	9.409e-1
10	rs7901695	6.563e-12	8.366e-12	2.009e-06	2.009e-06	2.009e-06	2.009e-06	1.005e-06	1.327e-0

```
  16   rs8050136  1.006e-08  1.192e-08   0.003078   0.003078   0.003074   0.003074   0.000778   0.0102
  16   rs3751812  1.017e-08  1.205e-08   0.003112   0.003112   0.003107   0.003107   0.000778   0.0102
  10   rs7904519  2.478e-08  2.913e-08   0.007586   0.007586   0.007558   0.007557   0.001423   0.0187
   3   rs7615580  2.789e-08  3.274e-08   0.008537   0.008536     0.0085     0.0085   0.001423   0.0187
  10   rs7903146  3.889e-08  4.551e-08     0.0119     0.0119    0.01183    0.01183   0.001435   0.0189
   3   rs6768587  3.966e-08  4.639e-08    0.01214    0.01214    0.01207    0.01207   0.001435   0.0189
   3   rs2028760   4.22e-08  4.934e-08    0.01292    0.01292    0.01283    0.01283   0.001435   0.0189
```

This table is sorted by significance, and the first SNP jumps out as being far and away the most significant. But to quickly talk about the data in the columns:

CHR and SNP contain the chromosome and SNP IDs, and UNADJ contains the unadjusted p-value that would have been present in the plain gwas_as.assoc file.

GC contains genomic-control corrected p-values, BONF contains Bonferroni single-step adjusted p-values, HOLM contains step-down adjusted p-values, SIDAK_SS contains Sidak single-step adjusted p-values, SIDAK_SD contains Sidak step-down adjusted p-values, FDR_BH contains Benjamini & Hochberg (1995) step-up FDR control, and FDR_BY contains Benjamini & Yekuteili (2001) step-up FDR control.

It's much beyond the scope of this paper to begin to compare the different methods of correction for multiple testing, but the important thing to note is that GC is very similar to UNADJ, and then the remaining tests result in very similar values as each other.

6i) How many SNPs show a significant (p<0.05) p-value under all of the multiple testing correction approaches?

We could probably just subset the table based on BONF values since it appears the strictest, but just for the sake of being strictly correct I'll subset based on what the question asks – that the p-values of all approaches are below 0.05.

```
adj_as <- read.csv('gwas_as.assoc.adjusted', header=T, sep='')
```

```
adj_sig <- adj_as[which(adj_as$UNADJ < 0.05 &
                        adj_as$GC < 0.05 &
                        adj_as$BONF < 0.05 &
                        adj_as$HOLM < 0.05 &
                        adj_as$SIDAK_SS < 0.05 &
                        adj_as$SIDAK_SD < 0.05 &
                        adj_as$FDR_BH < 0.05 &
                        adj_as$FDR_BY < 0.05),]
dim(adj_sig)
```

```
## [1]  9 10
```

Only nine SNPs have an adjusted p-value below 0.05 in all of the multiple testing correction approaches!

```
adj_sig
```

```
##   CHR       SNP     UNADJ        GC      BONF      HOLM  SIDAK_SS
## 1   3 rs6802898 2.327e-20 3.599e-20 7.123e-15 7.123e-15 7.123e-15
## 2  10 rs7901695 6.563e-12 8.366e-12 2.009e-06 2.009e-06 2.009e-06
## 3  16 rs8050136 1.006e-08 1.192e-08 3.078e-03 3.078e-03 3.074e-03
## 4  16 rs3751812 1.017e-08 1.205e-08 3.112e-03 3.112e-03 3.107e-03
## 5  10 rs7904519 2.478e-08 2.913e-08 7.586e-03 7.586e-03 7.558e-03
```

```
## 6   3 rs7615580 2.789e-08 3.274e-08 8.537e-03 8.536e-03 8.500e-03
## 7  10 rs7903146 3.889e-08 4.551e-08 1.190e-02 1.190e-02 1.183e-02
## 8   3 rs6768587 3.966e-08 4.639e-08 1.214e-02 1.214e-02 1.207e-02
## 9   3 rs2028760 4.220e-08 4.934e-08 1.292e-02 1.292e-02 1.283e-02
##    SIDAK_SD    FDR_BH    FDR_BY
## 1 7.123e-15 7.123e-15 9.409e-14
## 2 2.009e-06 1.005e-06 1.327e-05
## 3 3.074e-03 7.780e-04 1.028e-02
## 4 3.107e-03 7.780e-04 1.028e-02
## 5 7.557e-03 1.423e-03 1.879e-02
## 6 8.500e-03 1.423e-03 1.879e-02
## 7 1.183e-02 1.435e-03 1.896e-02
## 8 1.207e-02 1.435e-03 1.896e-02
## 9 1.283e-02 1.435e-03 1.896e-02
```

Again, one stands out in particular as being the most significant SNP: **rs6802898**.

Looking it up on NCBI tells us that it's an SNV in the PPARG (peroxisome proliferator activated receptor gamma) gene. This gene is one of the PPAR subfamily of nuclear receptors, which regulate transcription of various genes. It also notes that PPARG has been implicated in the pathology of numerous diseases including obesity, diabetes, atherosclerosis and cancer. It also looks like the SNV is located in the middle of an intron of the gene, so it's not immediately clear how the SNV modifies function.

The next most significant SNP, **rs7901695**, is a few orders of magnitude below the most significant, but a few orders of magnitude above what follows.

Looking this up on NCBI as well tells us that it's an SNV in the TCF7L2 (transcription factor 7 like 2) gene. This gene encodes a high-mobility group box-containing transcription factor, and plays a key role in the Wnt signaling pathway (a key cascade regulating development that is highly associated with cancer). Genetic variants of this gene have been found to be associated with increased risk of type 2 diabetes, and the protein itself has been implicated in blood glucose homeostasis. This SNV is also located in the middle of an intron of the gene.

6ii) is there evidence for population structure which may be confounding the analysis?

We can check by clustering, and visualize it with an MDS plot. We already have our .genome file from when we did relatedness QC, so we can use the following command:

```
plink --bfile gwas --read-genome gwas_genome.genome --cluster --mds-plot 10
```

Which produces 3 cluster files (gwas_cluster.cluster{1,2,3}), and an mds table (gwas_cluster.mds).

cluster1 contains information on the final solution, listed by cluster. So we can easily see how many clusters there are with wc -l:

```
wc -l gwas_cluster.cluster1
```

```
1
```

There's only 1 cluster. The fact that the individuals weren't classified into multiple clusters is a good indication that there isn't population structure confounding the data.

Cluster2 and cluster3 contain cluster data at the individual level, but since there's only 1 cluster with all of the individuals in it, it's not worth looking at them.

The mds file looks like the following:

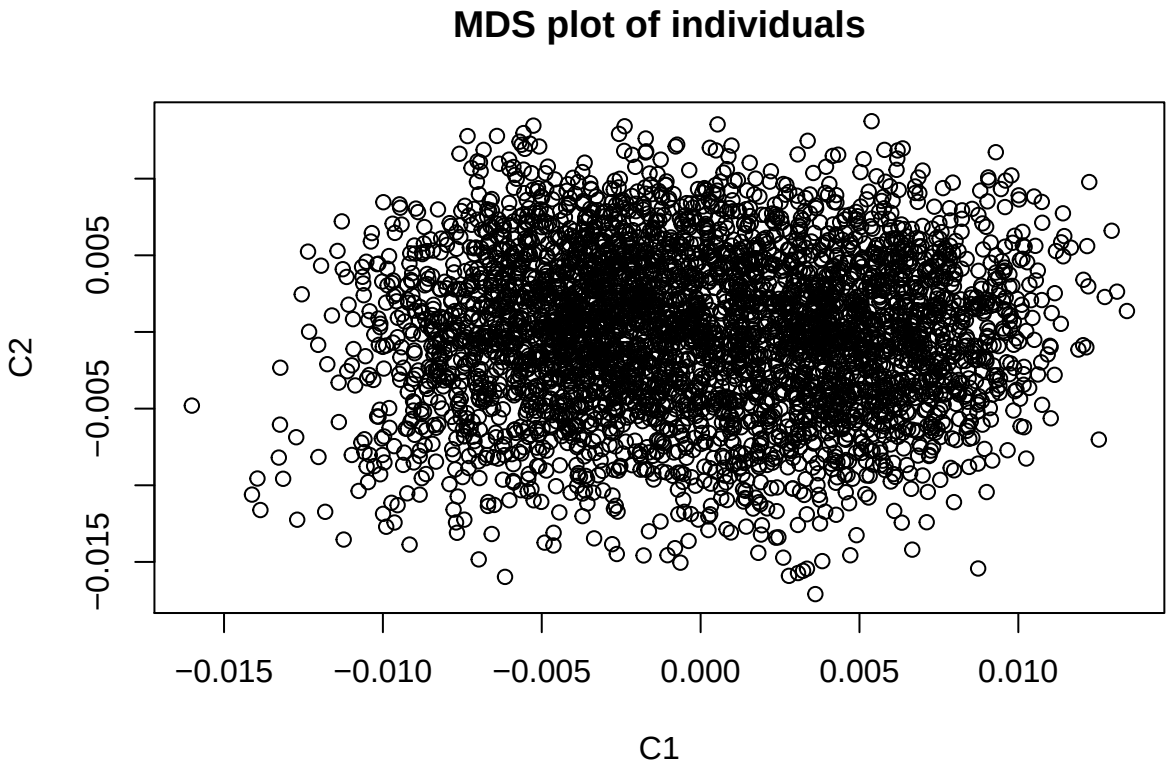
FID	IID	SOL	C1	C2	C3	C4	C5	C6	
0	A2001	0	-0.00318324	0.00161157	-0.00233766	0.00319896	-0.00488514	0.00104098	-0.00
1	A2002	0	-0.00353713	-0.00121527	-0.00211434	-0.0105191	0.00252363	0.0027438	-0.
2	A2003	0	-0.000185258	-0.00507529	0.0065482	0.0126053	-0.00116089	-0.00304759	0.0
3	A2004	0	-0.000856734	0.000521698	-0.00110272	0.00167534	-0.00353655	-0.00779197	0.0

Lots of columns, since we extracted 10 dimensions (the positions on which correspond to the columns C1-C10). The FID and IID contain the family and individual IDs as before, and SOL contains the assigned solution code (all lines should have 0, since this is obtained from –cluster and every individual was assigned to the same cluster).

We can plot just the first 2 dimensions to see a visualization of the clustering:

```
mds <- read.csv('gwas_cluster.mds', header=T, sep='')
```

```
plot(mds$C1, mds$C2, main='MDS plot of individuals',
     xlab='C1', ylab='C2')
```



As expected from when we saw that there weren't multiple clusters of individuals, we have one giant blob without any shape or direction. At this point it's safe to assume that there isn't population structure confounding the results.

7. Logistic regression

The last angle we can look at is to use the additional information of sex (contained in the PED file) and age (given as a covariate file) as covariates in a logistic regression.

This can be done with the following command:

```
plink --bfile gwas --logistic --sex --covar gwas.covar --adjust --out gwas_logit
```

This produces two files: gwas_logit.assoc.logistic, and gwas_logit.assoc.logistic.adjusted.

Looking first at gwas_logit.assoc.logistic:

```
head -n 5 gwas_logit.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
1	rs3934834	995669	T	ADD	3818	1.029	0.3812	0.7031
1	rs3934834	995669	T	AGE	3818	1.002	1.118	0.2635
1	rs3934834	995669	T	SEX	3818	1.012	0.1909	0.8486
1	rs3737728	1011278	A	ADD	3982	1.019	0.3867	0.699

CHR and SNP are the chromosome and SNP IDs, BP is the position on the Chromosome, and A1 is the minor allele.

TEST contains the tests on the SNP – ADD for the additive effects of allele dosage, AGE for the age covariate, and SEX for the sex covariate.

NMISS contains the number of non-missing individuals included in the analysis, OR contains the odds ratio, STAT contains the coefficient t-statistic, and P contains the p-value associated with the t-statistic.

If the p-values for the ADD tests remain high and the p-values for the covariate tests remain low, then it can be concluded that the allele does have an effect independent of the covariates.

Let's look at our most significant SNP from the previous analysis:

```
cat gwas_logit.assoc.logistic | grep rs6802898
```

3	rs6802898	12366207	T	ADD	3894	2.091	9.183	4.179e-20
3	rs6802898	12366207	T	AGE	3894	1.003	1.363	0.1729
3	rs6802898	12366207	T	SEX	3894	0.986	-0.217	0.8282

It appears that the ADD p-value is about the same as before, and the p-values of AGE and SEX aren't significant, even before correcting for multiple testing.

Moving on, let's look at gwas_logit.assoc.logistic.adjusted. It's just about identical to the previous adjusted p-value table, but now the p-values have the effects of the covariates removed.

Let's take a look at the top 9 and see if they're the same as the 9 from before:

```
head -n 10 gwas_logit.assoc.logistic.adjusted
```

CHR	SNP	UNADJ	GC	BONF	HOLM	SIDAK_SS	SIDAK_SD	FDR_BH	FDR_B
3	rs6802898	4.179e-20	6.73e-20	1.279e-14	1.279e-14	1.279e-14	1.279e-14	1.279e-14	1.69e-1
10	rs7901695	6.778e-12	8.865e-12	2.075e-06	2.075e-06	2.075e-06	2.075e-06	1.037e-06	1.37e-0
16	rs8050136	1.525e-08	1.833e-08	0.004667	0.004667	0.004656	0.004656	0.001248	0.0164
16	rs3751812	1.632e-08	1.961e-08	0.004996	0.004996	0.004983	0.004983	0.001248	0.0164
3	rs7615580	2.048e-08	2.454e-08	0.006269	0.006269	0.00625	0.00625	0.001248	0.0164
10	rs7904519	2.893e-08	3.453e-08	0.008854	0.008854	0.008815	0.008815	0.001248	0.0164
10	rs7903146	3.252e-08	3.878e-08	0.009955	0.009955	0.009906	0.009906	0.001248	0.0164
3	rs6768587	3.357e-08	4.002e-08	0.01028	0.01028	0.01022	0.01022	0.001248	0.0164
3	rs2028760	3.67e-08	4.371e-08	0.01123	0.01123	0.01117	0.01117	0.001248	0.0164

Though they're a fraction of a fraction less significant than before, these are the same SNPs with just about the same adjusted p-values. It seems that very little of what we saw in the previous analysis was confounded by age or sex, and rs6802898 is still incredibly significant.

8. Manhattan plot

"Use the R qqman package to produce a Manhattan plot visualizing the results of the association test from step 5 above."

I'd love to, but the output of step 5 (the –model command) doesn't contain all of the necessary data (notably, it doesn't have the base position on the chromosome).

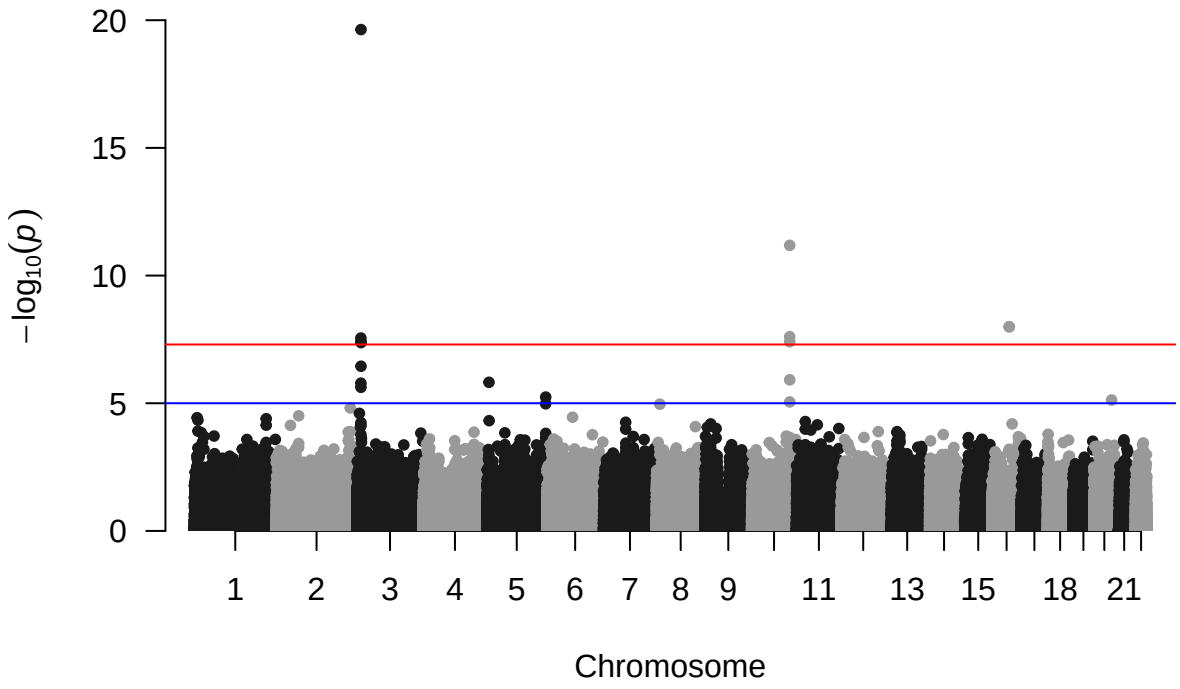
The only file that contains all of the data required to make a manhattan plot – CHR, SNP, BP, and P – is the gwas_as.assoc from the association test that didn't correct for multiple testing.

```
library(qqman)
```

```
##

## For example usage please run: vignette('qqman')

##

## Citation appreciated but not required:

## Turner, S.D. qqman: an R package for visualizing GWAS results using Q-Q and manhattan plots. biorXiv

##
```

```
gwas <- read.csv('gwas_as.assoc', header=T, sep='')
```

```
manhattan(x=gwas)
```



It works! And up at the top above Chr3 sits rs6802898, which indicates that this seems to have given us what we expected to see.