Genetic data concepts and tests

Cavan Reilly

September 4, 2019

Table of contents

Overview

Prior to conducting tests of association between markers and traits, it is common to examine the marker data for a couple of types of genetic correlation:

- linkage disequilibrium
- Hardy-Weinberg equilibrium

Linkage disequilibrium

Linkage disequilibrium exists when there is an association between the alleles that are found at 2 loci.

Linkage disequilibrium typically exists only for closely related loci and trails off at greater distances.

Moreover, it doesn't decay smoothly, rather there appear to be blocks of loci where all loci in the block are associated.

In association studies we use this property of genomes to identify locations that impact the trait under investigation.

If we can uncover loci that impact the trait we can then see if there are nearby genes and examine what is known about their functions.

Linkage disequilibrium

There are several methods in use for quantifying the extent of linkage disequilibrium between a pair of loci.

Suppose we have data on n subjects, and consider 2 sites where each site has 2 possible alleles (e.g. SNPs).

Suppose the alleles at site 1 are represented with a and A while the alleles at site 2 are represented with b and B.

Further suppose that the probability of observing these alleles is given by p_a, p_A, p_b and p_B.

If the alleles that occur at these 2 sites are independent then we can compute the probability of observing a genotype by just multiplying the probability of observing the alleles.

Linkage disequilibrium

In fact, consider the following table of expected counts if there is linkage equilibrium:

	B	b	
A	$n_{11} = Np_Ap_B$	$n_{12} = Np_Ap_b$	$n_{1.} = Np_A$
a	$n_{21} = Np_ap_B$	$n_{22} = Np_ap_b$	$n_{2.} = Np_a$
	$n_{.1} = Np_B$	$n_{.2} = Np_b$	$N = 2n$

Linkage disequilibrium

Compare this to the table of expected counts if there is linkage disequilibrium:

	B	b	
A	$n_{11} = N(p_A p_B + D)$	$n_{12} = N(p_A p_b - D)$	$n_{1.} = Np_A$
a	$n_{21} = N(p_a p_B - D)$	$n_{22} = N(p_a p_b + D)$	$n_{2.} = Np_a$
	$n_{.1} = Np_B$	$n_{.2} = Np_b$	$N = 2n$

where $D = p_{AB} - p_A p_B$. Clearly if $D = 0$ we get the previous table.

Note that in order to estimate D we need estimates of p_{AB}, p_A and p_B.

Linkage disequilibrium

While estimating p_A and p_B is straightforward, we can just use the sample proportions, $n_{1\cdot}/N$ and $n_{\cdot 1}/N$ respectively, estimating p_{AB} would seem to require knowledge of haplotypes which we don't have.

However we can estimate p_{AB} using the method of maximum likelihood.

Linkage disequilibrium

One problem with using D to measure the extent of linkage disequilibrium is that its upper bound depends on the allele frequencies, hence it is difficult to interpret.

For this reason we use

$$D' = \frac{|D|}{D_{\max}}$$

where

$$D_{\max} = \begin{cases} \min(p_A p_b, p_a p_B) & D > 0 \\ \min(p_A p_B, p_a p_b) & D < 0 \end{cases}$$

Values of D' near 1 indicate linkage disequilibrium while values near zero indicate linkage equilibrium.

Linkage disequilibrium

We will now see how to use R to compute estimates of D'.

First, load the genetics package and attach the fms data set.

```
> library(genetics)
> attach(fms)
```

Then we use the genotype function to create genotype objects from 2 of the SNPs in the ACTN3 gene as follows.

```
Actn3Snp1 <- genotype(actn3_r577x,sep="")
Actn3Snp2 <- genotype(actn3_rs540874,sep="")
```

Linkage disequilibrium

We need to specify `sep=""` as the default separator of alleles is the / symbol, and our data just has the alleles paired with no separator.

```
> actn3_r577x[1:10]
[1] CC CT CT CT CC CT TT CT CT CC
> Actn3Snp1[1:10]
 [1] "C/C" "C/T" "C/T" "C/T" "C/C" "C/T" "T/T" "C/T"
 [8] "C/T" "C/C"
Alleles:  C T
```

Linkage disequilibrium

Now the data are in a suitable format for computing LD using the LD function.

```
> LD(Actn3Snp1,Actn3Snp2)
Pairwise LD
-----------
                    D         D'       Corr
Estimates: 0.1945726 0.8858385 0.7860811
               X^2 P-value   N
LD Test: 895.9891         0 725
```

Linkage disequilibrium

If we now compare this to the situation where we look at SNPs in different genes, we see the level of LD is much lower.

As an example, we will use a SNP from the gene estrogen receptor 1.

```
Esr1Snp1 <- genotype(esr1_rs1801132,sep="")
> LD(Actn3Snp1,Esr1Snp1)
Pairwise LD
-----------
                      D         D'        Corr
Estimates: 0.01466542 0.1122922 0.06722353
               X^2    P-value   N
LD Test: 6.534478 0.01058033 723
```

So while the estimate is lower ($D' = 0.11$ instead of $D' = 0.89$) the p-value for the test indicates that it is different from 0 using conventional cutoffs.

Linkage disequilibrium

However, despite what the documentation for this package says, this is not a valid test for LD because the haplotype frequency is estimated and then used as if it was known.

The estimate of the haplotype frequency has uncertainty associated with it and this test ignores that uncertainty.

Also note that the sample size used for the test is based on the number of alleles, which is twice as large as the sample size.

This is not a valid value for the sample size unless we have Hardy-Weinberg equilibrium at both loci (more on this below).

Linkage disequilibrium

We can also examine LD between sets of SNPs. First we set up a few more SNPs from the ACTN3 gene.

```
Actn3Snp3 <- genotype(actn3_rs1815739,sep="")
Actn3Snp4 <- genotype(actn3_1671064,sep="")
Actn3AllSnps <- data.frame(Actn3Snp1,Actn3Snp2,Actn3Snp3,Actn3Snp4)
```

Linkage disequilibrium

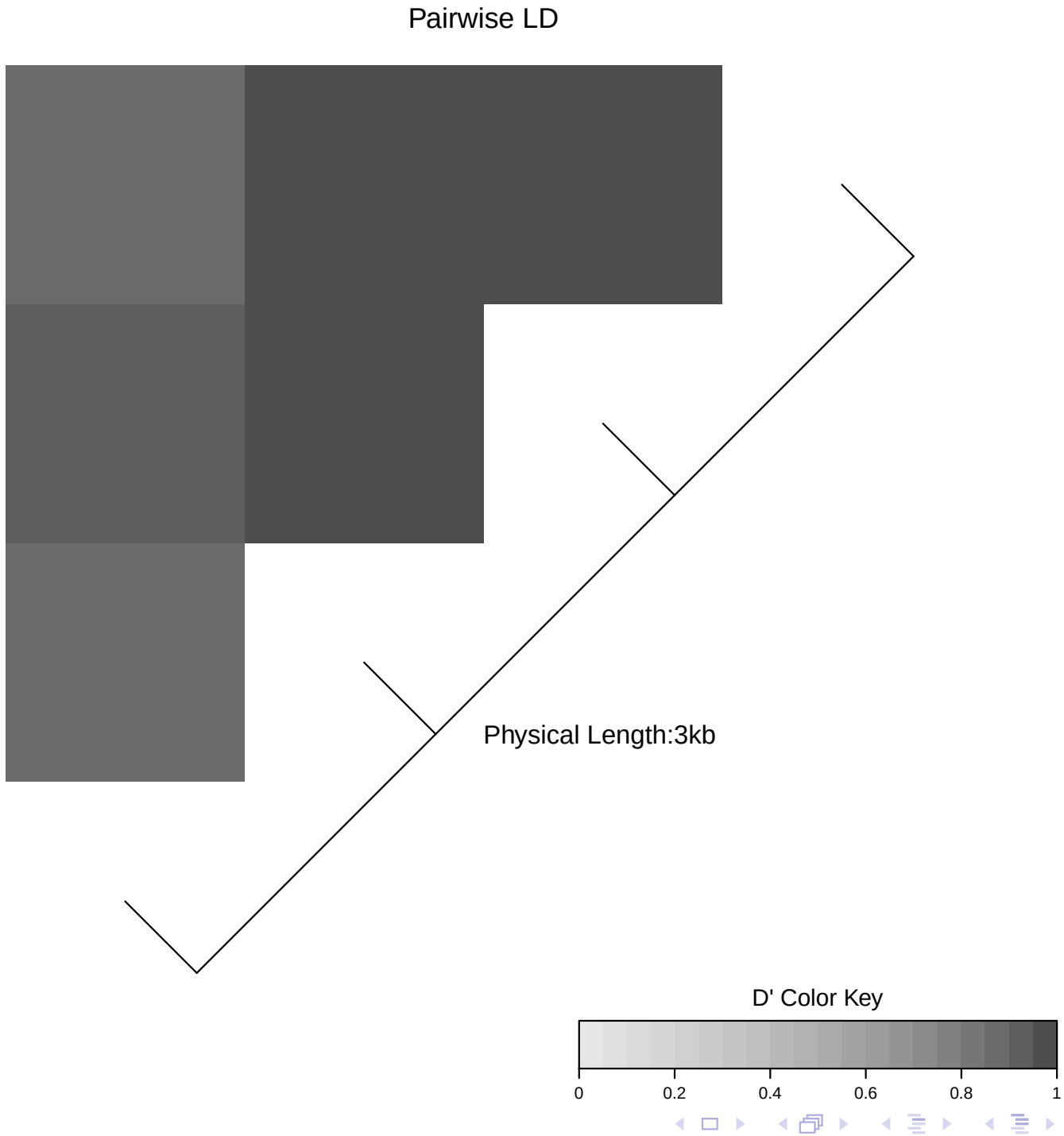
Here we will just pick out the estimates of D' from the output rather than examine all of the output.

```
> LD(Actn3AllSnps)$"D'"
           Actn3Snp1 Actn3Snp2 Actn3Snp3 Actn3Snp4
Actn3Snp1         NA 0.8858385 0.9266828 0.8932708
Actn3Snp2         NA        NA 0.9737162 0.9556019
Actn3Snp3         NA        NA        NA 0.9575870
Actn3Snp4         NA        NA        NA        NA
```

Linkage disequilibrium

We can also make a heatmap to graphically illustrate the extent of LD using the LDheatmap package.

```
> install.packages("LDheatmap")
> library(LDheatmap)
Loading required package:  grid
> LDheatmap(Actn3AllSnps,LDmeasure="D'")
```


Pairwise LD
Physical Length:3kb
D' Color Key
0
0.2
0.4
0.6
0.8
1

Linkage disequilibrium: r^2

Another way to measure LD is via a quantity related to the Pearson's χ^2 test statistic applied to the tables presented earlier.

This measure is $r^2 = \chi_1^2/N$, where χ_1^2 is the Pearson's χ^2 test statistic and $N = 2n$.

It can be shown that

$$r^2 = \frac{D^2}{p_A p_B p_a p_b}$$

so that there is a close connection between D' and r^2. Both adjust the measure D, they just do so in slightly different ways.

Linkage disequilibrium: r^2

We can use the same function to compute r^2, but specify `R^2` in place of `D'`. For example:

```
> LD(Actn3AllSnps)$"R^2"
           Actn3Snp1 Actn3Snp2 Actn3Snp3 Actn3Snp4
Actn3Snp1         NA 0.6179236 0.6729845 0.6375185
Actn3Snp2         NA        NA 0.9435869 0.9000219
Actn3Snp3         NA        NA        NA 0.8994410
Actn3Snp4         NA        NA        NA        NA
```

Linkage disequilibrium blocks

There appear to recombination hotspots in the genome: locations where recombinations frequently occur.

For regions between these hotspots there is high LD with groups of alleles segregating through populations.

To see these one can compute the average LD over regions by finding the mean level of LD over some region.

Average LD

This is easy to do in R, for example:

```
> LDMat <- LD(Actn3AllSnps)$"D'"
> mean(LDMat,na.rm=T)
[1] 0.9321162
```

A tagging SNP is a SNP selected from a LD block to represent all SNPs in that block.

The use of such SNPs reduces the problem of multiple testing, however sets of tagging SNPs seem to differ across populations, thus the use of tagging SNPs in association studies with outbred populations may require stratified analyses.

Hardy-Weinberg equilibrium (HWE)

Hardy-Weinberg equilibrium at a locus is said to exist when mating decisions are made without reference to alleles at that locus.

When HWE exists, genotype frequencies only depend on allele frequencies provided we ignore mutations and there is no immigration.

For genetic markers that are not associated with traits that impact fitness we expect HWE to hold, however if a certain allele gives individuals an advantage in terms of reproduction, then we would expect that the frequency of that allele to increase over time.

If a genetic locus does not obey HWE then allele frequencies depend on time, and combining data from different generations to estimate an "allele frequency" is problematic.

If there is immigration or noninterbreeding subgroups in a population, again, the notion of "allele frequency" is problematic.

HWE

We can use either Pearson's χ^2 tests or Fisher's exact test to test for HWE.

If our marker only has 2 alleles then the 2 categorical variables we conceptually have data about are which allele an individual has on the chromosome he received from his mother and which allele he has on the chromosome he received from his father.

With unrelated subjects we can't distinguish which allele came from which parent if someone is heterozygous, so we only observe 3 cells: if the 2 alleles are represented by a and A then the 3 cells are AA, Aa and aa.

HWE

If we use the following notation for genotype frequencies

AA	Aa	aa
n_{11}	n_{12}	n_{22}

then $\mathrm{E}n_{11} = Np_A^2$, $\mathrm{E}n_{12} = 2Np_A(1-p_A)$ and $\mathrm{E}n_{22} = N(1-p_A)^2$

but recall we can estimate p_A with $(2n_{11} + n_{12})/(2N)$ so that we can use the usual $(\text{observed} - \text{expected})^2/\text{expected}$ and sum over all cells in the table to get a χ^2 test statistic in the usual fashion.

Testing HWE in R

We can conduct these tests in R using the genetics package.

```
> attach(hgdp)
> Akt1Snp1 <- genotype(AKT1.C0756A,sep="")
> HWE.chisq(Akt1Snp1)
        Pearson's Chi-squared test with simulated
        p-value (based on 10000 replicates)
data:  tab
X-squared = 6.927, df = NA, p-value = 0.007699
```

So we conclude that this locus is not in HWE.

Testing HWE in R

However if we select a subset of the data we see that the results of the χ^2 test differ from the version of Fisher's exact test that is appropriate in this situation

```
> Akt1Snp1Maya <- genotype(AKT1.C0756A[Population=="Maya"],sep="")
> table(Akt1Snp1Maya)
Akt1Snp1Maya
A/A C/A C/C
  1   6  18
```

Testing HWE in R

```
> HWE.chisq(Akt1Snp1Maya)
        Pearson's Chi-squared test with simulated
        p-value (based on 10000 replicates)
data:  tab
X-squared = 0.287, df = NA, p-value = 1
> HWE.exact(Akt1Snp1Maya)
        Exact Test for Hardy-Weinberg Equilibrium
data:  Akt1Snp1Maya
N11 = 18, N12 = 6, N22 = 1, N1 = 42, N2 = 8, p-value
= 0.4843
```

However both fail to reject the null hypothesis that there is HWE at this locus.

HWE and population substructure

In the presence of

population admixture-interbreeding of 2 genetically distinct populations, or

population stratification-the existence of 2 or more genetically distinct groups that don't interbreed, HWE will not hold.

As such, failure of HWE can be used as a method to assess if either of these 2 phenomena are present.

Or it may be a sign of genotyping errors, in any event, it is a good thing to check prior to further analysis.

Genotyping errors

A genotyping error occurs when the genotype measured by the algorithm is not the underlying genotype.

As mentioned, one can test for HWE for each variant and then not consider variants that are not in HWE.

There are a number of problems with this approach:

- deviations from HWE could be due to associations between genotypes and disease status-a number of improvements have been suggested but these too are problematic
- departure from HWE could be due to population substructure
- multiple hypothesis testing-how do we interpret all of these tests.

Despite these difficulties, in practice SNPs are frequently dropped from the analysis due to a failure of HWE.

Identifying population substructure

While self-declared race can be useful for stratifying analyses, ethnicity operates on a much finer level than categories such as African-American.

With sufficient marker data one can detect trends in ethnicity as one traverses Europe.

Hence there is a demand for tools that can allow us to identify substructure and account for it in the analysis.

Once identified, one can either conduct a stratified analysis (i.e. analyze the different subgroups separately) or account for the subpopulations in a multiple regression type of approach.

Multidimensional scaling

Multidimensional scaling (MDS) relies on the idea that the collection of SNP data for each subject can be thought of as residing in a high dimensional space.

For example, think of the SNP data as the number of copies of the major allele for each SNP.

If there are 3 SNPs, then each subject has data of the form, e.g. $(0, 1, 0)$.

Now imagine these are the coordinates of a point in a 3 dimensional space.

So each subject is represented by a point in 3 dimensional space, and we can think about the distance between pairs of points in this space.

Multidimensional scaling

With many SNPs it is impossible to visualize these points but we can still define distances between individuals.

The goal of multidimensional scaling is to represent high dimensional data points in a lower dimensional space so that the pairwise distances between individuals are retained.

This would allow us to graphically assess if there are groups of subjects that are close in the high dimensional space, thereby allowing us to see if there is population substructure.

If we can identify such subpopulations, then we could define indicator variables for each subject that encode the subpopulation identity and include these in a multiple regression model to determine if a SNP impacts a trait given the effect of the subpopulation.

Principal components analysis

We can use principal components analysis (PCA) to much the same end.

The idea is that maybe all of the subjects collections of SNPs look like the SNP data for just a few subjects with small amounts of noise added.

We then try to identify the reduced sets of collections of SNPs and determine for each subject which of the few SNP collections to which he or she is closest.

Substructure identification in R

To use MDS, we first need to generate a table that has all of the pairwise distances between subjects based on the gene AKT1.

We will first extract all SNPs that cover this gene, then create a table that has the genotype for each subject encoded numerically then compute the distances between each subject.

```
> attach(fms)
> NamesAKT1Snps <- names(fms)[substr(names(fms),1,4)=="akt1"]
> NamesAKT1Snps
 [1] "akt1_t22932c"          "akt1_g15129a"          "akt1_g14803t"
 [4] "akt1_c10744t_c12886t" "akt1_t10726c_t12868c"
...
[22] "akt1_g22187a"          "akt1_a22889g"          "akt1_g23477a"
```

So there are 24 SNPs that intersect this gene.

Substructure identification in R

Now select the SNPs and convert to a numerical format (we need to give numeric values to the NAs or cmdscale will fail).

```
> FMSgeno <- fms[,is.element(names(fms),NamesAKT1Snps)]
> dim(FMSgeno)
[1] 1397 24
> FMSgenoNum <- data.matrix(FMSgeno)
> FMSgenoNum[is.na(FMSgenoNum)] <- 4
```

Then compare the 2 tables.

Substructure identification in R

```
> FMSgeno[1:5,1:4]
akt1_t22932c akt1_g15129a akt1_g14803t akt1_c10744t_c12886t
1            TT           GG           GG                    CC
2            TT           AA           TT                    CC
3            TT           AA           TT                    CC
4            TT           AG           GT                    CC
5            TT           AG           GT                    CC
> FMSgenoNum[1:5,1:4]
akt1_t22932c akt1_g15129a akt1_g14803t akt1_c10744t_c12886t
1             3            3            1                     1
2             3            1            3                     1
3             3            1            3                     1
4             3            2            2                     1
5             3            2            2                     1
```

Substructure identification in R

next compute distances.

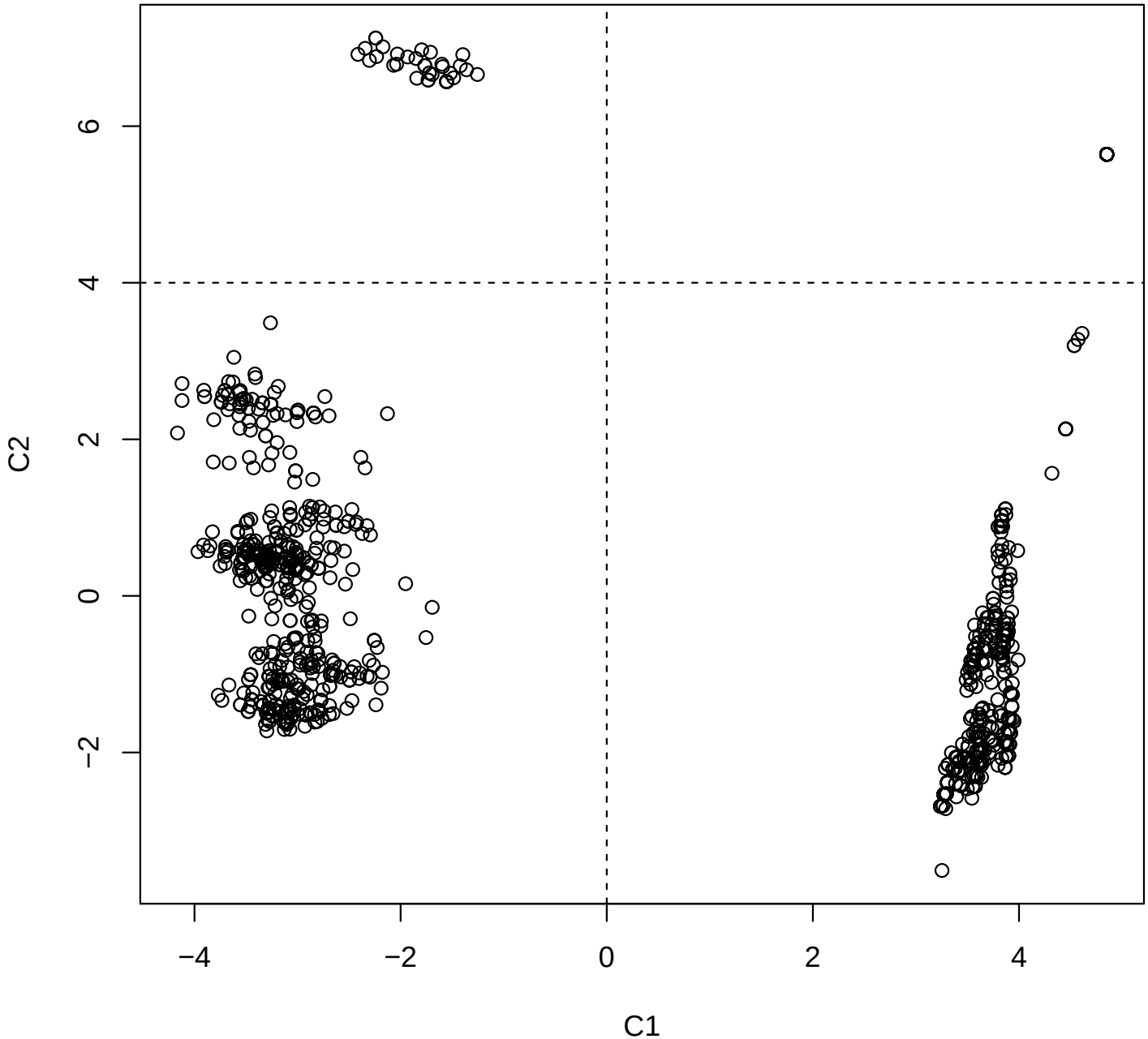
```
> DistFmsGeno <- as.matrix(dist(FMSgenoNum))
> DistFmsGeno[1:5,1:5]
      1        2        3        4        5
1 0.000000 4.795832 5.291503 3.741657 3.162278
2 4.795832 0.000000 2.236068 3.872983 3.000000
3 5.291503 2.236068 0.000000 3.162278 3.741657
4 3.741657 3.872983 3.162278 0.000000 2.449490
5 3.162278 3.000000 3.741657 2.449490 0.000000
```

Substructure identification in R

Finally generate a figure.

```
> plot(cmdscale(DistFmsGeno),xlab="C1",ylab="C2")
> abline(v=0,lty=2)
> abline(h=4,lty=2)
```

Some clusters are clear in the low dimensional representation of the data.

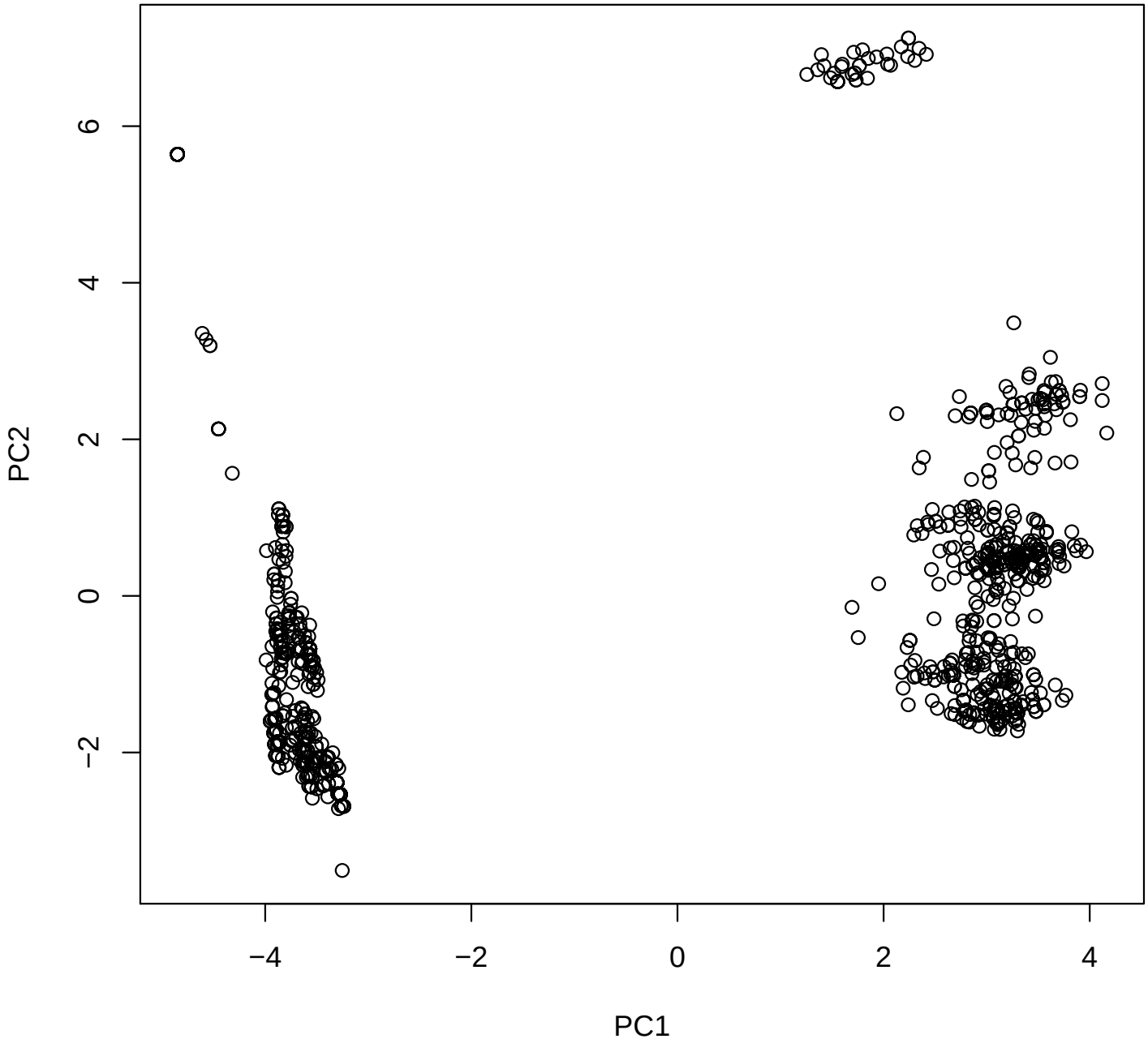

C2
6
4
2
0
−2
−4
−2
0
2
4
C1

Substructure identification in R

PCA is also straightforward in R, here we just plot the first 2.

```
PCFMS <- prcomp(FMSgenoNum)
plot(PCFMS$"x"[,1],PCFMS$"x"[,2],xlab="PC1",ylab="PC2")
```

Again a couple of distinct clusters are observed.


PC2
6
4
2
0
−2
−4
−2
0
2
4
PC1

Accounting for substructure

While the informal method described above involving multiple regression models using indicators identified using graphical tools is reasonable, how to deal with substructure is an ongoing problem.

One controversial aspect of trying to do this is that we use the same SNPs for identifying population substructure as we do for testing for associations.

Some of the main approaches are

- genomic control-use a different sampling distribution when testing for associations than the usual χ^2.
- structured association-much like our graphical approaches except use a formal model based clustering algorithm.
- eigenstrat-program that formalizes the PCA based graphical approach described above.