GWAS 5: Relatedness and population structure

Matti Pirinen, University of Helsinki

Updated: March 19, 2025.

This document is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. The slide set referred to in this document is "GWAS 5".

5.1 Genetic relatedness

All humans are genetically related to each other but the level of relatedness varies between pairs of individuals. In GWAS, we need to account for these differences in the levels of relatedness because otherwise they could bias the statistical association between genotypes and phenotypes. Here, we define what relatedness is and how it can be estimated. Later on the course we will discuss more why and how we can use estimates of relatedness and population structure in GWAS.

See slides (3-6) that summarize the inheritance process across generations that generates variation in the relatedness among humans or other diploid species.

A review of the relatedness estimates: Speed & Balding 2015, Relatedness in the post-genomic era: is it still useful?.

5.1.1 Defining relatedness

Consider two genomes G_1 and G_2 . Suppose that we could determine, for each genomic position x, the number of generations $t(x; G_1, G_2)$ that have passed since the most recent common ancestor (MRCA) genome of both G_1 and G_2 at site x was alive. From such information, we could define a relatedness value between G_1 and G_2 as the average time to MRCAs over some selected set of L loci. (Slide 5)

The relatedness measures used in practice try to capture a part of this perfect relatedness information, although not in terms of the number of generations since MRCA, but rather as a relative measure compared to the other pairs of genomes in the study population. This shortcut is made because estimating time to MRCA would require multiple assumptions about population genetic history, which are difficult to verify, and would involve demanding computation. Instead, some simple relatedness statistics used in GWAS can be easily computed for millions of pairs of individuals, and we will study these next.

Terms

- Identical-by-state (IBS). Two genomes are IBS at locus x if the DNA sequences of the genomes are identical at locus x. Here locus can be one or more nucleotides long. The observed IBS distribution for a pair of individuals is described by three values, (IBS₀, IBS₁, IBS₂), that tell which proportion of the genome between these individuals is of IBS 0,1 or 2, respectively. (Slide 7)
- Identical-by-descent (IBD). Two genomes are IBD(t) at locus x, if they have inherited identical DNA sequence at locus x from a common ancestor who has lived at most t generations ago. This is a well-defined concept, if pedigree information for a few generations is available. Typically, in GWAS data, no such information is available, and the concept of IBD is used without explicitly defining what

is the reference timeframe t. Then it is expected that a reasonable definition of the reference level is implicitly determined by the method that is used for estimating whether two genomes are IBD. Most often, IBD is defined by assuming that population allele frequencies are known and that there is IBD sharing between a pair of individuals, if the pair shares more allele IBS than would be expected for a random pair from the population. IBD implies IBS but not vice versa. (Slide 7.) The IBD distribution for a pair of individuals is described by three values, (IBD₀, IBD₁, IBD₂), that tell which proportion of the genome is estimated to be of IBD 0,1 or 2, respectively. The known pedigree relationships, such as full-sibs, determine only the **expected value** of IBD sharing; the realized IBD sharing has variation within any other relationship category except the parent-offspring pairs. (Slides 8-10).

- Kinship coefficient ϕ_{ij} is the probability that one allele sampled from individual *i* and one allele sampled from the same locus from individual *j* are IBD.
- Inbreeding coefficient $f_i = \phi_{MF} = 2\phi_{ii} 1$, where *M* and *F* are the mother and father of *i*. It describes how closely the two genomes of an individual *i* are related. $f_i = 0$ if the parents of *i* have nothing IBD. If f_i is large, the individual is said to be *inbred*. Outbred means not inbred.
- Relatedness coefficient $r_{ij} = 2\phi_{ij}$. Assuming no inbreeding, r_{ij} can be interpreted as the proportion of the genome that is shared between *i* and *j*.

In practical applications, make always clear whether you are considering kinship or relatedness coefficients as the values differ by a factor of 2. Here are some values for relatives.

If the purpose is to assign individuals into categories according to close relationships, a commonly-used inference procedure of the KING software is based on the powers of $\frac{1}{2}$. According to it, a pair is inferred to be a kth degree relative pair based on the following cut points for the estimated r_{ij} or ϕ_{ij} values:

degree	r_{ij} within	ϕ_{ij} within
mono twins	$> \frac{1}{2^{0.5}} = 0.707$	$> \frac{1}{2^{1.5}} = 0.354$
1st deg.	$\left(\frac{1}{2^{1.5}}, \frac{1}{2^{0.5}}\right) = (0.354, 0.707)$	$\left(\frac{1}{2^{2.5}}, \frac{1}{2^{1.5}}\right) = (0.177, 0.354)$
2nd deg.	$\left(\frac{1}{2^{2.5}}, \frac{1}{2^{1.5}}\right) = (0.177, 0.354)$	$\left(\frac{-1}{2^{3.5}}, \frac{-1}{2^{2.5}}\right) = (0.088, 0.177)$
3rd deg.	$\left(\frac{1}{2^{3.5}}, \frac{1}{2^{2.5}}\right) = (0.088, 0.177)$	$\left(\frac{1}{2^{4.5}}, \frac{1}{2^{3.5}}\right) = (0.044, 0.088)$
unrelated	$<\frac{1}{2^{3.5}}=0.088$	$<\frac{1}{2^{4.5}}=0.044$

Note that the expected value of relatedness for each category is the lower bound multiplied by $\sqrt{2} = 2^{0.5}$ and is simultaneously also the upper bound divided by $\sqrt{2}$.

5.1.2 Estimating relatedness

Example 5.1. Let's generate genotype data for 5 pairs of full siblings, 5 pairs of half siblings and 10 unrelated individuals from one population using p = 10,000 SNPs whose MAF range from 0.2 to 0.5. We will then demonstrate different relatedness estimates on these data by showing the 30x30 relatedness matrix of the individuals.

Let's put the within family data generation into its own function that allows generating pairs of offspring that share 2 parents (full sibs), 1 parent (half sibs) or 0 parent (unrelated).

```
offspring.geno <- function(n.families, n.snps, fs = rep(0.5, n.snps), n.shared.parents = 2){
   #INPUT:</pre>
```

```
# n.families, number of families where each family produces two offspring (>0)
```

```
# n.snps, number of independent SNPs used in simulation (>0)
```

- # fs, vector of allele 1 freqs for SNPs, length == n.snps, values in (0,1)
- # n.shared.parents, 0,1 or 2 shared parents for the two offspring in each family

```
#OUTPUT:
# X, the genotypes of (2*n.families) offspring,
     is matrix of dimension (2*n. families) \times n. snps and with values in \{0, 1, 2\}
stopifnot(n.families > 0)
stopifnot(n.snps > 0)
stopifnot(all(fs > 0 & fs < 1) & length(fs) == n.snps)</pre>
stopifnot(n.shared.parents %in% 0:2)
if (n.shared.parents == 2) parents = list(c(1,2), c(1,2)) #parents [[1]] are the parents of offspring 1
if(n.shared.parents == 1) parents = list(c(1,2), c(3,2))
if(n.shared.parents == 0) parents = list(c(1,2), c(3,4))
n.parents = 4 - n.shared.parents #4, 3 or 2 for values of n.shared.parent of 0, 1 or 2
X = matrix(0, nrow = 2*n.families, ncol = n.snps)
for(ii in 1:n.families){ #each "family" means a pair of offspring that share 'n.shared.parents'
  x.parents = t(replicate(2*n.parents, rbinom(n.snps, size = 1, prob = fs))) #2*n.parents parental g
  for(offs in 1:2){ #for two offspring within "family"
    #phase is the indicator of whether offs inherits each parents' 1st allele or not
    phase = t(replicate(2, rbinom(n.snps, size = 1, prob = 0.5))) #phase has one row for each parent
      for(i.parent in 1:2){
        for(ph in 0:1){
          loci = (phase[i.parent,] == ph) #which loci from i.parent have phase ph?
          #add to current offs' genotype i.parent's allele from the correct phase
          X[2*(ii-1) + offs, loci] =
            X[2*(ii-1) + offs, loci] + x.parents[2*parents[[offs]][i.parent] - ph, loci]
        }
      }
  }
}
return (X)
```

And now the simulation of the genotype matrix X for 30 individuals and 10,000 SNPs.

}

IBS estimator It is straightforward to count the proportion of loci where a pair of individuals share 0,1 or 2 alleles IBS. These IBS values already make a distinction between different levels of relatedness and a plot of, say, IBS0 against IBS2 shows often a clear structure. However, the absolute amount of IBS sharing depends heavily on the allele frequencies in the population and methods that estimate kinship or relatedness coefficients typically aim for estimating IBD.

We generate each IBSz-matrix by multiplication over possible sharing patterns: There are three possible ways to have IBS2: either sharing genotype 2 or genotype 1 or genotype 0. And there are two ways to be IBS1: one individual must be heterozygous while the other must be one of the homozygotes. Finally, the overall IBS estimate is IBS2 + 0.5×10^{-5} mesons.





Avg. IBS from 10000 SNPs

The numerical values of this IBS plot depend on the population allele frequencies and are not as easily mapped to the known relationship categories as the IBD based measures for which, for example, $r \sim 50\%$ for full sibs and $r \sim 25\%$ for half sibs. Additionally, it seems that separation of the half sibs from the unrelated in the IBS matrix is not very clear.

Correlation estimator (GCTA's genetic relationship matrix) Let's look at how a standard measure of correlation applied to a large number p of variants can be interpreted as an estimate of IBD.

Consider a locus l where frequency of allele 1 is f_l and denote by $A_{hl} \in \{0, 1\}$ the random variable describing the allele of genome h at locus l. Under the assumption that all genomes in the sample come from the same homogeneous population, we have that $E(A_{hl}) = f_l$ and $Var(A_{hl}) = f_l(1 - f_l)$ for all h. Denote by r_{hk} the (unknown) probability that genomes h and k are IBD at a locus. We have

$$\Pr(A_{hl} = 1 = A_{kl} \mid f_l, r_{hk}) = r_{hk} f_l + (1 - r_{hk}) f_l^2,$$

where the first term describes the case where h and k are IBD at l (with probability r_{hk}) and happen to share the same copy of allele 1 (with probability f_l), and the second term describes the case where h and k are not IBD at l (prob. $1 - r_{hk}$) and happen to carry independent copies of allele 1 (prob. f_l^2). We can solve for

$$r_{hk} = \frac{\Pr(A_{hl} = 1 = A_{kl}) - f_l^2}{f_l(1 - f_l)} = \frac{\mathbb{E}(A_{hl}A_{kl}) - \mathbb{E}(A_{hl})\mathbb{E}(A_{kl})}{\sqrt{\operatorname{Var}(A_{hl})\operatorname{Var}(A_{kl})}} = \operatorname{cor}(A_{hl}, A_{kl})$$

Thus, for genomes, their IBD proportion can be estimated by the correlation of their allele states.

Let $G_{il} = A_{i_1l} + A_{i_2l} \in \{0, 1, 2\}$ be the genotype of individual *i* at locus *l* and assume HWE: $Var(G_{il}) = 2f_l(1 - f_l)$.

$$\operatorname{cor}(G_{il},G_{jl}) = \frac{\operatorname{Cov}(A_{i_1l} + A_{i_2l},A_{j_1l} + A_{j_2l})}{2f_l(1-f_l)} = \frac{(r_{i_1j_1} + r_{i_1j_2} + r_{i_2j_1} + r_{i_2j_2})f_l(1-f_l)}{2f_l(1-f_l)} = \frac{4\phi_{ij}}{2} = 2\phi_{ij} = r_{ij}.$$

Thus, also at the level of the genotypes, correlation can be used as an estimate of the IBD sharing between the individuals.

We use all available SNPs (after some pruning that we'll talk about later) to estimate the pairwise correlations using the formula

$$\widehat{r_{ij}} = \frac{1}{p} \sum_{l=1}^{p} \widehat{\text{cor}}(G_{il}, G_{jl}) = \frac{1}{p} \sum_{l=1}^{p} \frac{\left(g_{il} - 2\,\widehat{f}_l\right) \left(g_{jl} - 2\,\widehat{f}_l\right)}{2\,\widehat{f}_l \left(1 - \widehat{f}_l\right)},$$

where $g_{il} \in \{0, 1, 2\}$ is the observed genotype of i and \hat{f}_l is the allele 1 frequency estimate from the sample, both at locus l.

For n individuals, these pairwise relatedness coefficients form an $n \times n$ genetic relationship matrix (GRM) for which we use notation **GRM-cor** to emphasise that it is the correlation based GRM. This GRM-cor has been widely used through the popular linear mixed models software GCTA. Also PLINK has an efficient implementation of computing GRM-cor.

The interpretation of negative values of $\hat{r_{ij}}$ is that those pairs are less related than an expected pair of unrelated (no-IBD sharing) individuals. This is possible, for example, if there is population structure in the data because then the individuals from different populations can look less related than the theoretical construction of a pair of "unrelated individuals" based on the allele frequency data from the combined sample. A related, and possibly more severe, problem is that when the sample is not from a single homogeneous population, then individuals from the same population seem more related than they actually are (for example 3rd degree relatives can look like 2nd degree relatives). Thus, the correlation estimator, or any other estimator that defines the reference level of IBD based on the sample's allele frequencies, is not robust to population structure.

Note also a possible numerical instability of the estimator at SNPs where MAF is close to 0. Then the denominator $\hat{f}_l(1-\hat{f}_l) \approx 0$, and therefore a pair of carriers of the rare allele will get a huge positive contribution to their relatedness estimate from the rare allele. For example, if $\hat{f}_l = 0.001$, then a pair of individuals who share the minor allele will get a huge contribution of $(1-0.002)^2/(2 \cdot 0.001 \cdot 0.999) = +498.5$ to their relatedness estimate, whereas individuals who share the common genotype 0, will get only a tiny contribution of +0.002 and a pair who have different genotypes 0 and 1 will get contribution of only -1.00. Intuitively, it feels correct to assume that individuals who share a rare allele indeed are likely to have recent IBD sharing at that locus but the unboundedness of the estimator with rare alleles seems a technical problem that leads to a huge variance of the estimator. Therefore, we typically restrict the use of the GRM-cor estimator to common variants (say MAF>5%).

Let's make GRM-cor for our current data set.

```
X.scaled = scale(X) #standardize each SNP at columns of X
GRM = (X.scaled %*% t(X.scaled))/p #correlation matrix of individuals based on standardized SNPs
```

GRM from 10000 SNPs



Let's print out which pairs seem to be full sibs, $0.4 < r_{ij} < 0.6$ or half sibs, $0.15 < r_{ij} < 0.35$.

full.sibs = which(GRM > 0.4 & GRM < 0.6, arr.ind = TRUE) #each pair is twice here since R is symmetric full.sibs = full.sibs[full.sibs[,1] < full.sibs[,2],] #pick only pairs where 1st index < 2nd index rbind(full = as.vector(t(full.sibs)))

```
      ##
      [,1]
      [,2]
      [,3]
      [,4]
      [,5]
      [,6]
      [,7]
      [,8]
      [,9]
      [,10]

      ##
      full
      21
      22
      23
      24
      25
      26
      27
      28
      29
      30
```

```
half.sibs = which( GRM > 0.15 & GRM < 0.35, arr.ind = TRUE)
half.sibs = half.sibs[half.sibs[,1] < half.sibs[,2], ]
rbind(half = as.vector(t(half.sibs)))
```

[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] ## half 11 12 13 14 15 16 17 18 19 20

Seems correct.

PLINK's $\hat{\pi}$ ("**pi-hat**") A widely used IBD estimate is $\hat{\pi}$ of the PLINK software. It was developed by Purcell et al. (2007) (p.565-566).

PLINK uses a method-of-moments approach to estimate the probability of sharing 0, 1, or 2 alleles IBD for any two individuals from the same homogeneous, random-mating population. It is based on the fact that at any one locus, by the law of total probability, for k = 0, 1, 2,

$$\Pr(IBS = k) = \Pr(IBS = k | IBD = 0) P(IBD = 0) + \Pr(IBS = k | IBD = 1) P(IBD = 1) + \Pr(IBS = k | IBD = 2) P(IBD = 2) P(IBD$$

This is a system of three equations (k = 0, 1, 2), and if we substitute the left hand side with the observed IBS status and if we can estimate the conditional probabilities Pr(IBS = k | IBD = z) for all $k, z \in \{0, 1, 2\}$, then we can solve for the three unknowns IBDz = P(IBD = z), z = 0, 1, 2.

Assuming a homogeneous population with known population allele frequencies, it is straightforward to derive the IBS probabilities given the IBD state using the same logic as with the correlation estimate above (i.e., the alleles can be IBS either because they are IBD or because they are not IBD but they are sampled from the population frequencies and happen to be IBS). Thus, the conditional probabilities can be computed for each locus given the allele frequency estimates at that locus.

Finally, PLINK averages the equations across the loci resulting in three equations where IBSk on the left hand side is the proportion of IBS= k across the genome, and the conditional probabilities of (IBDk | IBSz) are also averages across the genome. Then IBDz can be solved from these three equations for z = 0, 1, 2.

PLINK uses these IBD estimates to compute, for each pair i and j,

$$\widehat{\pi}_{ij} = \frac{1}{2} \text{IBD1}_{ij} + \text{IBD2}_{ij}.$$

It is an estimate of r_{ii} , that is, the proportion of genome shared IBD.

Since $\hat{\pi}_{ij}$ is based on the allele frequencies from the sample, its interpretation as a measure of IBD sharing has the same problems as the correlation estimate. In particular, if the population is not homogeneous, the values are biased compared to the expectations under the known relationship categories.

KING The IBD relatedness estimators explained above assume that the individuals come from a homogeneous population. KING by Manichaikul et al. 2010 is a relatedness estimation method that is more robust to population structure and is very efficiently implemented. It derives a kinship estimate for a pair of individuals without reference to the population allele frequencies. Instead, it gets the relatedness information from the difference between the counts of loci where both individuals are heterozygotes $(N_{1,1})$ and counts of loci where they are different homozygotes $(N_{2,0})$, normalized by the sum of the heterozygous loci of the individuals:

$$\widehat{\phi_{ij}} = \frac{N_{1,1} - 2N_{2,0}}{N_1^{(i)} + N_1^{(j)}}.$$

Let's apply the KING estimator to the current dataset with the full sibs, half sibs and unrelated, and let's multiply the KING's kinship estimate by 2 to get from the kinship scale to the scale of the relatedness coefficient r.

KING (r) from 10000 SNPs



Looks correct. We'll later come back to the difference between KING and GRM-cor when there is some population structure in the data.

KING was recently used to estimate relatedness for all 1.2e+11 pairs from the UK Biobank data (~500,000 samples)(Slide 12).

IBD segments Genome is inherited in chunks and therefore individuals sharing an allele IBD at a locus are also likely to share a larger region IBD around that locus. Most accurate relatedness estimates come from methods that can model this spatial structure of IBD segments. Those methods are usually computationally intensive and not applicable to GWAS data sets with 10^4 or 10^5 individuals and we don't study them on this course. A study by Ramstetter et al. 2017 lists and compare many of those methods.

5.1.3 Use of IBD estimates in quality control

(From Purcell et al. 2007.)

IBD sharing estimates can be used for quality control (QC) and to indicate and diagnose errors in records or genotype data, including sample swaps, sample duplications, and sample contamination events, as well as misspecified or undetected familial relationships. For example, values of IBD2 near 1 indicate either duplicated samples or monozygotic twins.

If DNA from some individual(s) contaminates other samples, this can lead to a distinctive pattern of contaminated samples showing high IBD with all other individuals. This is because contamination induces false heterozygote calls (e.g., AA pooled with CC may well be genotyped as AC), and heterozygotes cannot ever be IBS=0 with any other SNP genotype, which artificially inflates the IBD estimates. Furthermore, the contaminated samples will show strong, negative inbreeding coefficients, indicating more heterozygote loci than expected. This is why excessive heterozygocity is a QC criterion in GWAS data.

See also slides 16-17 for an application of relatedness estimates in QC.

5.1.4 Uses of relatedness estimates outside GWAS

Several companies offer direct-to-customer service to analyze customer's DNA and over 30 million people have already subscribed to such services globally. One of the uses is to reveal genetic relatives based on the estimated IBD sharing among the customers. The purpose of uniting distant relatives is nice, but the power of genetics can also cause complicated situations related to unexpected family trees as also described in this article "With genetic testing, I gave my parents the gift of divorce".

The rapid growth of the public databases, where voluntary individuals have uploaded their genetic data and other information (such as age and location), have already been used to identify from a database a relative of an external DNA sample (e.g. from a crime scene), which has led to a complete identification of the external DNA sample. A famous example is the case of the Golden State Killer in CA, US, where in April 2018 a suspect was arrested for murders and rapes committed in 1970s and 1980s. The police got the lead by uploading the DNA found from a crime scene to a GEDmatch database and identified 10-20 3rd degree cousins whose additional information helped to identify the suspect. How lucky were the police? by Edge and Coop. A Science paper by Erlich et al. 2018 summarizes this approach and its profound future prospects (Slides 13-15):

"Consumer genomics databases have reached the scale of millions of individuals. Recently, law enforcement authorities have exploited some of these databases to identify suspects via distant familial relatives. Using genomic data of 1.28 million individuals tested with consumer genomics, we investigated the power of this technique. We project that about 60% of the searches for individuals of European-descent will result in a third cousin or closer match, which can allow their identification using demographic identi Moreover, the technique could implicate nearly any US-individual of European-descent in the near future We demonstrate that the technique can also identify research participants of a public sequencing projec Based on these results, we propose a potential mitigation strategy and policy implications to human subject research."

5.2 Population structure

Several relatedness estimation methods considered above were base on the assumption of a *homogeneous* population, i.e., a population without *genetic structure*. Genetic population structure is present in the sample, if the sample can be divided into groups in such a way that individuals from one group are, on average, more genetically similar among themselves than with the individuals from the other groups.

5.2.1 Sources of population structure

Most common variants among the humans are shared across the world in the sense that those common alleles are present all around the world. Still, even the common variants carry information about the geographic location of their carrier since their allele frequencies differ across the world. (Slides 19-20.)

Example 5.2. Genetic drift Suppose that a population with 100 individuals is living in a particular area. Suppose that half of the population migrates to a new area and the two subpopulations do not have any contact with each other anymore, and in particular, do not have any genetic exhange in the following generations. Let's visualize how the allele frequency of a particular variant evolves in the two populations as a function of generations they have been separated. The assumption is that at time 0 the allele frequency is the same in both populations, and, in each of the following generations, the offspring alleles are randomly sampled from the existing alleles with replacement (one individual can have none, one or multiple offspring). This sampling means that Hardy-Weinberg equilibrium holds in each subpopulation. There are 50 individuals in each population, so 100 alleles in each population. The population size is assumed constant.

```
n = 100 #alleles in each generation of each subpopulation
f0 = 0.5 #starting allele frequency
K = 100 #number of generations in simulation
npop = 10 #Let's make 10 rather than just 2 pops to illustrate more general behavior
f = matrix(NA, ncol = K+1, nrow = npop) #results of allele freqs across pops and generations
for(pop in 1:npop){
    a = rbinom(n,1,f0) #starting configuration of alleles
    f[pop, 1] = mean(a) #allele frequency at generation 0
    for(ii in 1:K){
        a = sample(a, size = n, replace = TRUE) #resample generation ii
        f[pop, 1 + ii] = mean(a) #allele frequency at generation ii (index ii+1 since generation 0 is at in
    }
}
plot(NULL, xlab = "generation", ylab = "allele frequency",
        main = paste("n =",n), xlim = c(0, K), ylim = c(0, 1))
```



for(pop in 1:npop){lines(0:K, f[pop,], lwd = 1.4, col = topo.colors(npop)[pop])}

n = 100

generation

The figure shows ten possible evolutions for our subpopulations. We see that each population starts near the common allele frequency value (here 0.5) but, as time goes by, some populations differ strongly in their allele frequency. If we were to genotype individuals from generation 80 at this variant, we could tell, for example, that heterozygotes are definitely from one of the 5 populations not yet fixed to either allele and we could tell which are the possibilities for each homozygote individual. Thus, pure random variation in allele frequencies between generations generates marked allele frequency differences between populations over time. This is called **genetic drift**.

The amount of genetic drift is strongly dependent on the population size. Statistician might think that each new generation tries to "estimate" its parents' generation allele frequency by sampling alleles from the parents' generation and by computing their frequencies, and the precision of this estimate decreases as the sample size decreases. Let's repeat similar analysis but with 10 times larger population size. (And let's not repeat the code in the output, just the updated Figure.)



n = 1000

Within the same time span, the larger populations were much less affected by genetic drift. This phenomenon also explains why some human populations are more strongly diverged genetically from their neighbours than some other, geographically equally distant populations. The Finnish genetic ancestry is an example of a population with a relatively strong genetic isolation among the European populations, to a large part due to a relatively small historical "founder population" size in Finland. Consequently, the overall genetic background among individuals with Finnish genetic ancestry is less heterogeneous than in many other European countries (which make some genetic analyses simpler in Finland), and due to a strong historical genetic drift effect in Finland, some functional alleles have survived in Finland

with much higher frequency than elsewhere (say, e.g., 1% in Finland and 0.01% elsewhere) even though selection effect might be against them. We already know what a huge difference such a frequency difference has in terms of statistical power to discover the phenotype associations! These are reasons why international research community has put much focus on Finland when it comes to doing genetics research.

Above we saw that already a single variant shows informative genetic population structure between population groups that have been separated for a while. And when we combine information across 100,000s of variants across the genome, we can expect to see quite a clear structure, if we just can find suitable tools to pick it up.

5.2.2 Principal component analysis (PCA)

Suppose we have an $n \times p$ matrix **X** of genotypes of n individuals measured on p SNPs. We would like to summarize the main structure of these data with respect to the individuals by using only much less than p dimensions. Such **dimension reduction** task is most intuitive, if it results in only one or two dimensions since then we can easily visualize the results by simply plotting the individuals in their new coordinates on a line or a plane.

Imagine each individual being originally as a point in the p-dimensional space where each dimension corresponds to one SNP and the individual's coordinate on the axis is the genotype at that SNP. If we draw one

line through the *p*-dimensional space and project each individual on that line, then we can represent each individual by using only one value, the individual's coordinate on the chosen line. Now each individual is represented by one value instead of the original p values so we have completed a dimension reduction from p dimensions to 1 dimension. Is this useful? Only if we can choose that one line in such a way that it will capture a useful amount of the information in the data. In principal component analysis (PCA), first defined in 1901 by Karl Pearson, our criterion to choose the line is that the individuals' projections on the line should have **the largest variance possible** among all possible lines that we could draw through the p-dimensional space.

Let's make an example data of 2 SNPs where we have 50 individuals from two populations. In the blue population, the allele 1 frequencies are 5% and 95% at the two SNPs, and in the red population they are the opposite: 95% and 5%. We don't expect that either SNP alone would completely separate the two populations but maybe PCA is able to combine the information into one dimension in a way that shows the population structure from the data better than either of the SNPs alone. Let's draw a picture.

```
set.seed(20)
npop = 2
cols = c("blue","red")
f = matrix(c(0.05, 0.95, 0.95, 0.05), byrow = TRUE, ncol = npop)
p = nrow(f) #number of SNPs
n = rep(50, npop) #number of samples from each population
pop = rep(1:npop, n) #from which population each individual comes
X = c() #empty genotype data matrix
for(ii in 1:p){
  x = c() #empty genotype vector for SNP ii
  for(jj in 1:npop){
    x = c(x, rbinom(n[jj], size = 2, f[ii,jj])) #add genotypes of pop jj to x
  }
  X = cbind(X,x) #add SNP x as a new column to genotype matrix X
}
jitt.1 = runif(n[1], -0.03, 0.03) #add some noise to coordinates in plot to avoid overlap
jitt.2 = runif(n[2], -0.03, 0.03)
plot(X[,1] + jitt.1, X[,2] + jitt.2,
     col = cols[pop], xlab = "SNP1", ylab = "SNP2")
```



We have clear patterns in the original two dimensional SNP space where the individuals from the red population are mainly at the lower right corner and from the blue at the upper left corner. But neither SNP alone can separate the two populations, and if we colored all dots black we would not see a clear separation of the points in to two populations on this plot. Let's see which is the line on which the projections of these points have the largest variance. We get than from PCA computed by prcomp().

```
X = scale(X) #always standardize each variant before PCA
pca = prcomp(X) #do PCA; we look later what this function returns
plot(X[,1] + jitt.1, X[,2] + jitt.2, asp = 1, #Plot the points, now after scaling
        col = cols[pop], xlab = "SNP1 (stnd'd)",
        ylab = "SNP2 (stnd'd)")
abline(a = 0, b = pca$rotation[2,1]/pca$rotation[1,1]) #add the PC1 line
```



If we now project all points on the line determine by PC1, as has been done in vector pca\$x[,1], and use just random y-coordinates for visualization, we have:



The direction picked by the PC1 also happens to separate the two populations. So even with only two SNPs, neither of which alone separates the populations, PCA can combine their information to capture the main structure of the data, which here matches the existence of the two populations.

Although the 2-SNP-example above is just a toy demonstration, it gives us a good reason to expect that

when we use 100,000s of SNPs, the PCs, i.e., the directions of the largest genetic variation in the data, will capture population structure, if such exists.

Terms:

- **Principal components** of the genetic data are the linear one dimensional subspaces of the original genotype space that have the following properties: The first PC explains the maximum variance possible by any linear one dimensional subspace. For any k > 1, the k^{th} PC explains the maximum variance possible by any linear one dimensional subspace conditional on being orthogonal to all preceding PCs. The number of PCs is min $\{n, p\}$.
- Scores or principal component scores are the coordinates of the individuals when they have been projected on the PC. They are computed using the PC loadings and the individuals' (standardized) genotypes. In the object returned by prcomp(), the scores are in matrix x, so, e.g., the scores on PC 6 are in vector pca\$x[,6].
- Loadings are the coefficients that determine how the scores are computed from the (standardized) genotypes. Each PC k has a loading vector $\mathbf{l}_k = (l_{k1}, \ldots, l_{kp})^{\mathsf{T}}$ and the score of individual i on PC k is

$$\operatorname{score}_{ik} = \mathbf{l}_k^{\mathsf{T}} \mathbf{x}_i = \sum_{m=1}^p l_{km} x_{im},$$

where x_{im} is the standardized SNP genotype of *i* at SNP *m*. Note that when we have the loadings at hand, we can project also any external individual on the existing PCs generated by our reference data set. Loadings from prcomp() object pca are in matrix pca\$rotation.

Example 5.3. Let's generate data from three poulations P1,P2,P3 of which P1 and P2 are more closely related with each other and more distant from P3. A standard measure of differentiation is the Fst value that describes how large a proportion of genetic variation is present between populations compared to within populations. For example, we have Fst ~ 0.003 between the ancestries of Eastern and Western Finland and ~ 0.10 between genetic ancestries of different continents. The Balding-Nichols model to generate allele frequencies for two populations with Fst value of F first picks a background allele frequency f, for example, from a Uniform distribution, and then samples the subpopulation allele frequencies as independent draws from the distribution

Beta
$$\left(\frac{1-F}{F}f, \frac{1-F}{F}(1-f)\right)$$
.

Let's generate data so that Fst between P1 and P2 is 0.003 and Fst between P3 and the shared ancestral population of P1 and P2 is 0.05. (This is a level of differentiation that we might observe between Eastern (P1) and Western Finnish ancestries (P2) and Northern African ancestry (P3).) Let's sample n = 100 individuals from each population using p = 3000 SNPs and show PCs 1-2.

```
n = 100 #per population
p = 3000 #SNPs
fst.12 = 0.003
fst.12.3 = 0.05
f = runif(p, 0.1, 0.5) #common SNPs in background population
f.3 = rbeta(p, (1-fst.12.3)/fst.12.3*f, (1-fst.12.3)/fst.12.3*(1-f))
f.12 = rbeta(p, (1-fst.12.3)/fst.12.3*f, (1-fst.12.3)/fst.12.3*(1-f)) #P1&P2's shared ancestral populat
f.1 = rbeta(p, (1-fst.12)/fst.12*f.12, (1-fst.12)/fst.12*(1-f.12))
f.2 = rbeta(p, (1-fst.12)/fst.12*f.12, (1-fst.12)/fst.12*(1-f.12))
f.2 = rbeta(p, (1-fst.12)/fst.12*f.12, (1-fst.12)/fst.12*(1-f.12))
#Let's check that f.1 and f.2 looks similar compared to f.1 and f.3 or f.2 and f.3
par(mfrow = c(1,3))
plot(f.1,f.2, main = paste("Fst", fst.12), xlim = c(0,1), ylim = c(0,1), pch = 3)
plot(f.1,f.3, main = paste("Fst >", fst.12.3), xlim = c(0,1), ylim = c(0,1), pch = 3)
```



Let's then generate the genotype data and do PCA.

```
x = cbind(
    replicate(n, rbinom(p, size = 2, p = f.1)), #generate n inds from P1
    replicate(n, rbinom(p, size = 2, p = f.2)), # from P2
    replicate(n, rbinom(p, size = 2, p = f.3))) # from P3
x = t(x) #each replicate (=ind) is now in a column, but we want inds to rows and SNPs to cols
pca = prcomp(x, scale = TRUE) #do PCA
cols = rep( c("cyan", "skyblue", "magenta"), each = n) #color for each ind according to pop
plot(pca$x[,1], pca$x[,2], col = cols, xlab = "PC1", ylab = "PC2")
```



We see that the first PC picks P3 apart from P1 & P2 but does not separate P1 and P2. The second PC starts to separate P1 and P2. The separation between P1 and P2 would become more clear if we added a few thousand more SNPs or if we increased the Fst value between P1 and P2 (left as an exercise).

Example 5.4. Let's do PCA with real allele frequency data from the 1000 Genomes project. Let's read in a frequency file (that is based on the files from IMPUTE2 webpage) and see what's in it.

```
af = read.table("http://www.mv.helsinki.fi/home/mjxpirin/GWAS_course/material/afreq_1000G_phase1_chr15-
as.is = TRUE, header = TRUE)
dim(af)
```

[1] 5266 19

af[<mark>1</mark>,]

id position a0 a1 ASW CEU CHB CHS CLM FIN GBR chr## 1 15 rs11248847 20101049 G A 0.2377 0.1882 0.4278 0.335 0.1917 0.1613 0.1461 ## IBS JPT LWK MXL PUR TSI YRI ## 1 0.2143 0.3876 0.2165 0.25 0.2091 0.1888 0.09659

We have allele 1 frequency info for 5,266 SNPs in 14 populations. The SNPs have been chosen from chromosomes 15-22 and are at least 100,000 bps apart to avoid highly correlated SNPs. Additionally, the global MAFs of all these SNPs are > 5% but some of them may very rare in some of the populations.

- ASW [AFR] (61) African Ancestry in Southwest US
- CEU [EUR] (85) Utah residents (CEPH) with Northern and Western European ancestry
- CHB [ASN] (97) Han Chinese in Beijing, China
- CHS [ASN] (100) Southern Han Chinese
- CLM [AMR] (60) Colombian in Medellin, Colombia
- FIN [EUR] (93) Finnish from Finland
- GBR [EUR] (89) British from England and Scotland
- IBS [EUR] (14) Iberian population in Spain
- JPT [ASN] (89) Japanese in Toyko, Japan
- LWK [AFR] (97) Luhya in Webuye, Kenya
- MXL [AMR] (66) Mexican Ancestry in Los Angeles, CA
- PUR [AMR] (55) Puerto Rican in Puerto Rico
- TSI [EUR] (98) Toscani in Italia
- YRI [AFR] (88) Yoruba in Ibadan, Nigeria

Note that sample sizes for some populations (IBS in particular) is small so we won't use them. Let's demonstrate a European PCA using GBR, TSI, CEU and FIN. We simply simulate some number of individuals (n = 50) from each population and do PCA. (We could use the original individual level 1000 Genomes data as well, but since that requires large files, here we just work with the allele frequencies and simulate our individuals from them.)

```
p = nrow(af) #number of SNPs
n = 50 #samples per population
pop.labs = c("GBR", "TSI", "CEU", "FIN")
pop = rep(1:length(pop.labs), each = n)
x = c()
for(ii in 1:length(pop.labs)){
    x = rbind(x, t(replicate(n, rbinom(p, size = 2, prob = af[,pop.labs[ii]]) ) ) )
}
x = x[, apply(x, 2, var) > 0 ] #keep only SNPs that have variation in data
pca = prcomp(x, scale = TRUE)
plot(pca$x[,1], pca$x[,2], col = pop, pch = 19, xlab = "PC1", ylab= "PC2",
    main = "Simulation from 1000 Genomes Phase 1")
legend("topright", leg = pop.labs, col = 1:length(pop.labs), pch = 19, cex = 1)
```



Simulation from 1000 Genomes Phase 1

It looks like the PC1 is determined by the North-South direction and the PC2 separates Central European and British ancestry from the Finnish and Italian ancestries.

plot(pca\$x[,1], pca\$x[,3], col = pop, pch = 19, xlab = "PC1", ylab= "PC3", main = "Simulation from 1000 Genomes Phase 1")



Simulation from 1000 Genomes Phase 1

18

And the PC3 then separates GBR and CEU.

Let's see some colorful pictures that PCA has produced with real data (slides 21-25).

How does PCA work? Technically, PCA is the eigenvalue decomposition of the correlation based genetic relatedness matrix (GRM-cor) that we discussed earlier: the eigenvectors of the GRM-cor are the scores on the PCs (first eigenvector corresponds to first PC etc.) This means that the GRM-cor can be seen as being built up from the PCs, one by one, as shown on slide 27.

Visualization of PCA.

Tutorial on PCA by Jon Shlens is an excellent tutorial, but unfortunately it uses the rows and columns of data matrix the other way as we are using so the data matrix is given as $p \times n$ matrix, and p (the number of variables) is called m.

How to carry out PCA? Most GWAS software packages can do PCA. Typically, the steps are as were done by Kerminen 2015:

- Identify a suitable set of individuals by excluding one individual from each pair of closely related individuals. (Close relatives can drive some leading PCs and hence mix up the analysis when the purpose is to find population structure, see an example below.)
- Identify a suitable set of SNPs, typically MAF>5% and strict LD-pruning is applied (we'll talk about pruning in the next section). Remove also the known regions of high-LD (Price et al. 2008).
- Make sure that the method is using mean-centered genotypes; often the SNPs are further satandardized to have variance 1 in the sample. Note: Also the GRM-cor uses the standardized genotypes as it is based on the genotype correlation.
- Plot the loadings along the genome to observe if there are some genomic regions with much larger contributions than the genome average. Spikes somewhere in the genome indicate that those regions are driving the PC, and that is most likely because of inadequate pruning of the SNPs from that region (See Figure 9 in Kerminen 2015).
- Visualize the PCs, for example by plotting two PCs against each other. Observe if there are outliers that seem to drive some of the leading (say first 20) PCs, and possibly remove those outliers, and redo PCA. Such individuals could have some quality problems or they may have genetic ancestry that is different from the rest of the sample, which can be a problem in a GWAS as we'll discuss soon. If you have geographic location info about your samples, color each PC-plot with the expected geographic populations of the samples. If the PC plot seems to separate the expected populations, it is likely to be correctly done, otherwise there is need to recheck the steps.
- Project the excluded relatives on the PCs. Some relatives may have been excluded while generating the PCs, but they will have valid scores on the PCs when projected using the loadings that were computed when those relatives were excluded.

Example 5.5. Why to exclude relatives from PCA? Let's make a data set that has 10 individuals from each of populations 1 and 2. The Fst between populations is 0.01 and the data includes one pair of full sibs from population 1, while the other individuals are unrelated within each population. Let's see how the first and the second PC behave.

```
set.seed(20)
p = 10000 #SNPs
fst.12 = 0.01
cols = c("black","limegreen")
```

```
pca = prcomp(X, scale = T) #do PCA
```

```
plot(pca$x[,1], pca$x[,2], asp = 1, col = cols[pop], xlab = "PC1", ylab = "PC2", pch = 19)
```





Thus, the first PC picks the relative pair and not the population structure. Only the second PC picks the population structure. This is not what we want in GWAS, where we want to adjust for relatedness in other ways and use PCA to get an idea of the population structure. Let's do the PCA by first removing one of the sibs, and then projecting him/her back among the others in the PC plot.

```
X.unrel = X[-1,] #remove row 1
pca = prcomp(X.unrel, scale = TRUE)
plot(pca$x[,1], pca$x[,2], asp = 1, col = cols[pop[-1]], xlab = "PC1", ylab = "PC2",
    main = "Unrelated PCA", pch = 19)
#project back individual 1 to the PCs spanned by the 19 other individuals
pca.1 = predict(pca, newdata = matrix(X[1,], nrow = 1)) #projects newdata to PCs defined in 'pca'
points(pca.1[1], pca.1[2], col = "violet", pch = 19, cex = 1.2) #ind 1 on PCs
```

Unrelated PCA



PC1

We see that the excluded sibling (in violet) is projected among his/her population, and now the PCs are not affected by any close relationships.

Note that when projecting samples using predict() on an object from prcomp(), the new data must be in the same units as the original data that was used to create the PCA with prcomp(). If the original data for prcomp() were as unscaled genotypes, then the new data must also be given as unscaled genotypes with the same allele coding. And if the input data were scaled outside the prcomp() function call, then the new data must also be scaled by using the same mean and SD values before the projection using the predict() function.

5.2.3 Relatedness estimates with population structure Let's see what happens when we have both population structure and **cryptic relatedness**, i.e., close relationships among the study participants that are unknown to us prior to accessing the genetic data.

Suppose we have two populations separated by an Fst of 0.1. We collect 4 pairs of half siblings from population 1 and 1 pair of half siblings from population 2. Let's use p = 10000 SNPs similar to our earlier sibship simulations.

GRM = (X.scaled %*% t(X.scaled))/p #correlation matrix of individuals based on standardized SNPs
#make KING-robust estimate for r:

Let's plot the results by dividing the relatedness coefficients into categories of monozygotic twins, 1st, 2nd, 3rd degree relatives and unrelated. Let's also print the relatedness values for five pairs of half sibs, who should belong to the category of the second degree relatives.

[1] 0.10459466 0.09525122 0.09165326 0.09614190 0.29005345



GRM from 9832 SNPs

We see that only the pair from Pop2 has its r_{ij} estimate corresponding to a value expected for the second deggree relatives. The other pairs have less GRM-cor relatedness in this sample because the strong population difference make them look like they were sharing more IBD than they actually do within their own population.

What about KING? (Suppressing code.)

[1] 0.2458297 0.2454545 0.2482046 0.2511605 0.2443132



KING (r) from 9832 SNPs

KING gets the half-sibs correct and r_{ij} values are around 25% as expected. It indeed seems robust to this kind of a population structure.

5.2.4 Fine-scale genetic structure Those interested in level of fine-scale genetic structure that can be extracted from genetic data can see results for

- Bantu-speaking African populations (Fortes-Lima et al. 2023)
- Central Asia (Mezzavilla et al. 2014)
- Estonia (Pankratov et al. 2020).
- Finland (Kerminen et al. 2017).
- France (Saint Pierre et al. 2020).
- Ireland (Gilbert et al. 2017).
- Japan (Sakaue et al. 2020)
- Netherlands (Byrne et al. 2020).
- Scotland (Gilbert et al. 2019)

- South and Southeast Asia (Tagore et al. 2021)
- Spain (Bycroft et al. 2019).
- UK (Leslie et al. Nature 2015).