BIOL 350: Bioinformatics

William Letsou

2024-04-04

Contents

Chapter 1

Introduction

Welcome to BIOL-350 (Spring 2024) at New York Tech! For this set of lectures, follow along with the slides here. Each chapter describes the theory and practice of the step in an analysis pipeline for you to conduct your own genetic association study.

Chapter 2

Principal Components Analysis

This week we'll see how individuals can be separated by genetic ancestry using principal components analysis. We'll practice applying PCA on a subset of the 1KGP data.

2.1 Pricipal components analysis: theory

The populations in our dataset can be separated into clusters based on their genotypes. The inferred groups help control for confounding due to ancestry, and are also more reliable than self-reported race in association studies. To see how it works, suppose $\mathbf{X}$ is an $n \times m$ (standardized) genotype matrix with individuals down the rows and SNPs across the columns. Principal components analysis (PCA) says we can find an $m \times n$ matrix $\mathbf{V}$, a diagonal $n \times n$ matrix $\mathbf{\Sigma}$, and an $n \times n$ matrix $\mathbf{U}$ satisfying

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{1}$$

If we think of $\mathbf{V}$ as the the (standardized) SNP genotypes of an "ideal" person of a certain ancestry, then

$$x_{j\cdot}v_{\cdot i} = u_{ji}\lambda_{jj} \tag{2.1}$$

represents the amount of idealized person i in actual person j, up to some proportionality constant λ_{jj} that depends on the ancestry. Then rows j of $\mathbf{U}$ are the *ancestries* of person j, and columns i of $\mathbf{U}$ are the ancestries of each individual on ancestry i. It is important to remember that these "idealized" ancestries do not necessarily correspond with our preconceived notions of ancestry, so we cannot interpret them as "European," "African," or "Asian," say. If we rearrange Eq. (1) and use the fact that the columns of $\mathbf{U}$ and $\mathbf{V}$ are *orthonormal*, we can find that

$$\mathbf{X}\mathbf{X}^T\mathbf{U} = \mathbf{U}\mathbf{\Sigma}^2, \tag{2.2}$$

meaning that the columns of **U** are the eigenvectors of the matrix

$$\frac{1}{m}\mathbf{X}\mathbf{X}^T \tag{2}$$

whose (i, j) entry is the genetic correlation between individuals i and j, sometimes known as the *genomic relationship matrix* or GRM. PCA works by finding the first several eigenvectors of the GRM and plotting each individual's ancestry along each orthogonal vector in a rectangular grid. Clusters of individuals in this grid represent distinct ancestry groups.

2.2 Principal components analysis: practice

2.2.1 Importing the data

To do PCA in R, we'll need to load the libraries SNPRelate and SeqArray:

```
library(SNPRelate)
library(SeqArray)
```

Download the chr1 vcf file containing just the CEU, YRI, and CHB populations. Once you have the file, store its name as a variable:

```
vcf <- "path/to/file.vcf.gz"
```

Now we'll convert the vcf format to gds format, retaining the base filename and changing the vcf.gz extension to gds. (This may take a minute to complete.) Then we'll import the gds file as a gds object.

```
seqVCF2GDS(vcf.fn = vcf,"path/to/file.gds") # convert vcf to gds with a new file name
genofile <- seqOpen("path/to/file.gds") # import the gds object
```

You can can see the various fields under genofile by printing it. To access the data in one of the fields, do

```
seqGetData(genofile,"sample.id") # view the sample ids
```

where the name of the field is enclosed in quotes.

2.2.2 Running PCA

To run PCA in R, simply do

```
pca <- snpgdsPCA(genofile) # runs PCA
```

to create an objects with 32 eignevectors of the GRM. Make a data frame of the first several eigenvectors along with subject ids:

```
df.pca <- data.frame(sample = pca$sample.id,EV1 = pca$eigenvect[,1],EV2 = pca$eigenvec
```

We'll plot individuals along EV1, EV2, and EV3 in several two-dimensional projections. But we'll want to see how the clustering done by PCA corresponds to individuals' self-reported race; for that we'll need another column in our data frame.

2.2.3 Getting population labels

The indivs file contains each subject id along with its 1KGP population group. Let's import it now:

```
indivs <- read.table("path/to/CHB+YRI+CEU.txt",header = FALSE)
colnames(indivs) <- c("id","pop")
```

The second field of this table is pop, an assignment to each id of one of three 1KG population groups. We want to match the right ID in indivs to the right ID in df.pca so that we can color our PCA plots by population. If the tables are in the same order, matching will be easy, but it not, we have to use the match(x,y) function, which finds the positions in y corresponding to the same items in x. Thus we can make a new column pop in df.pca with the corresponding pop values from indivs by

```
df.pca$pop[match(indivs$id,df.pca$sample)] <- indivs$pop # find the population group of each indi
```

2.2.4 Plotting

Now that we have a column of population labels, we can make a scatter plot colored by treating the pop column as vector of factors; we can get the unique values of a factor vector by applying the function levels() to it. A plot of the second principal component vs. the first can then be generated by

```
par(mar = c(5.1,5.1,4.1,2.1) )
plot(df.pca$EV1,df.pca$EV2,pch = 19,col = factor(df.pca$pop),xlab = "PC1",ylab = "PC2",cex.lab =
legend("topright",legend = levels(factor(df.pca$pop)),bty = "n",pch = 19,col = factor(levels(fact
```

Move the legend around if it covers any points, and make similar plots for the other two comparisons between the first three PCs.

Finally, close the connection to the gds file when you are done:

```
seqClose(genofile)
```

2.3 To turn in:

Make three (nicely formatted) plots of:

1. PC2 vs. PC1
2. PC3 vs. PC1
3. PC3 vs. PC2

For each plot, discuss:

1. Whether the populations appear to be well separated in PCA space
2. What the gradients of the different PCs represent, that is, what axis of variation each PC appears to explain
3. How to subset your df.pca data frame to isolate individuals of each population (i.e., provide code)

Chapter 3

Kinship Analysis

3.1 Kinship: theory

Kinship can be defined as the expected fraction of alleles that two individuals got from the same ancestor(s). We say that two individuals share an allele of a SNP *identical-by-descent* or *IBD* if they inherited the same copy of the allele from a common ancestor. IBD-sharing is different from simply carrying the same allele of a gene (known as *identical-by-state* or *IBS*-sharing), which unrelated individuals may do if the allele is common enough in the population. The *degree* R of relationship may be defined as the effective number of meioses separating the relatives through the equation

$$\frac{1}{2^R} = \frac{1}{2^{R_1}} + \frac{1}{2^{R_2}}, \tag{3.1}$$

in which R_i is the number of meioses separating the relatives through the first relative's ith parent. For example, sibs are connected by two meioses through two parents, while a parent and child are connected by one meiosis through one parent: both relationships are degree-1.

The probability that two relatives share an allele IBD is $\frac{1}{2^R}$, as there is a $\frac{1}{2}$ probability that an allele is passed on in any meiosis, and R is the effective number of meioses or steps between the relatives. If $2 \times \frac{1}{2^R}$ is the expected number of alleles shared IBD at any given locus, then the fraction of the genome shared by any two relatives is

$$r = \frac{2 \times \frac{1}{2^R}}{2} = \frac{1}{2^R}. \tag{3.2}$$

However, genomic sharing can be realized in different ways depending on the probabilities π_0, π_1, and π_2 that individuals share zero, one, or two copies IBD at a locus. The probability that the relatives inherit both both copies IBD, viz.,

$$\pi_2 = P\left(\text{share 2 IBD}\right) = 2^2 \frac{1}{2^{R_1} 2^{R_2}}, \tag{3.3}$$

is simply the product of the probabilities of sharing through both parents. The probability of sharing exactly one allele IBD,, viz.,

$$\pi_1 = P\left(\text{share 1 IBD}\right) = 2\left(\frac{1}{2^{R_1}} + \frac{1}{2^{R_2}}\right) - 2^3 \frac{1}{2^{R_1} 2^{R_2}}, \tag{3.4}$$

is the got by finding the probability $2\left(\frac{1}{2^{R_1}} + \frac{1}{2^{R_2}}\right) - 2^2 \frac{1}{2^{R_1} 2^{R_2}}$ of sharing at least one allele IBD less the probability π_2 of sharing two. Finally, the probability of sharing at zero alleles IBD, viz.,

$$\pi_0 = P\left(\text{share 0 IBD}\right) = 1 - 2\frac{1}{2^{R_1}} - 2\frac{1}{2^{R_2}} + 2^2 \frac{1}{2^{R_1} 2^{R_2}}, \tag{3.5}$$

is got by subtracting the probability $\pi_1 + \pi_2$ of sharing at least one allele IBD from 1. The coefficients account for the fact that there are 2 alleles at each locus and 2^2 that can be shared.

From (3.3)–(3.5), the fraction of the genome shared IBD is

$$r = \frac{2\pi_2 + 1\pi_1}{2} = \frac{1}{2^{R_1}} + \frac{1}{2^{R_2}} = \frac{1}{2^R}. \tag{3.6}$$

But the same value of r can obtain from different values of π_1 and π_2. For example, full sibs have a 25% probability of sharing two alleles and a 50% chance of sharing one allele at a locus, for a total fraction $r = 0.5$ shared. But a parent and child have 0% chance of sharing two alleles and a 100% chance of sharing one, also giving $r = 0.5$. Thus, your parent does is not equal your sibling, despite the fact of your sharing equal amounts of your genome with each of them. Put another way, parents cannot pass on their genotypes to their offspring.

3.1.1 KING

KING [1] computes both the probability π_0 that two relatives share 0 alleles IBD as well as the coefficient of relatedness $\phi = \frac{r}{2}$, defined as the probability that two alleles taken one from each relative are IBD at a locus (the maximum probability is $\frac{1}{2}$ because there is a 50% chance that the alleles chosen come from different parents). The idea is to compare the counts X and Y of the alternative alleles which two individuals each have at a genetic locus. If the pair are from a single ancestral population, the expected values and variances of the allele counts are $\mathbb{E}\left(X\right) = \mathbb{E}\left(Y\right) = 2p$ and $\sigma_X^2 = \mathbb{E}\left(X^2\right) - \mathbb{E}\left(X\right)^2 = \mathbb{E}\left(Y^2\right) - \mathbb{E}\left(Y\right)^2 = 2p\left(1-p\right)$. Thus the expected value of the difference $\mathbb{E}\left(\left(X-Y\right)^2\right) = \mathbb{E}\left(X^2\right) + \mathbb{E}\left(Y^2\right) - 2\mathbb{E}\left(XY\right)$ is

$$\frac{\mathbb{E}\left(\left(X-Y\right)^2\right)}{\sigma_X^2 + \sigma_Y^2} = 1 - \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = 1 - r, \tag{3.7}$$

where $\sigma_{XY} = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$ is the covariance of the genotype counts and $r_{ij} = 2\phi_{ij}$ is the genetic correlation between individuals i and j; the latter can be interpreted as the amount of the genome shared IBD.

KING estimates ϕ from Eq. (3.7) by counting the number N of loci at which two individuals are heterozygous Aa, Aa or opposite homozygous AA, aa, as well as the total number of alleles at which each individual is heterozygous Aa:

$$\hat{\phi_{ij}} = \frac{N_{Aa,Aa} - 2N_{AA,aa}}{N_{Aa}^{(i)} + N_{Aa}^{(j)}}. \tag{3.8}$$

From (3.8) it can be seen that shared heterozygous sites increase the estimated relatedness, and that unshared homozygous sites decrease relatedness. Eq. (3.8) is called a “robust” estimator because it measures relatedness in a purely pairwise fashion: it does not rely on population estimates of allele frequencies. However, if the individuals are not of the same genetic background, the allele frequency p is not well-defined and Eq. (3.7) does not hold, leading to negative estimates of ϕ; this feature is not necessarily a problem, as it helps us to distinguish different ancestries within a single population.

3.1.2 PC-AiR

KING provides an estimate of relatedness. But as cryptic relatedness can skew ancestry estimates determined by PCA, we should use kinship to correct PCA. This is where principal components analysis in related samples, or PC-AiR [2] comes in handy. PC-AiR identifies a subset of $\mathcal{U}$ of individuals who are unrelated to each other (via negative KING kinship estimates), but maximally related to individuals in a remaining set $\mathcal{R}$. Recalling the math in Week 1, the matrix

$$\frac{1}{m}\mathbf{W} = \mathbf{X}^T\mathbf{U} = \mathbf{V}\mathbf{\Sigma}, \tag{3.9}$$

so that $\frac{1}{m}w_{jk}\lambda_{kk}^{-1} = x_{ij}u_{ik}\lambda_{kk}^{-1} = v_{jk}$ is the genotype at SNP k in ancestry j inferred from subjects' genotypes $\mathbb{X}$ who are in subset $\mathcal{U}$. Then if we have an $m \times n_u$ genotype matrix $\mathbf{X}'$ of individuals in the subset $\mathcal{R}$, we can “apply” $\mathbf{W}$ to get an estimate of the “ancestry-corrected” PCs by:

$$\frac{1}{m}\mathbf{X}'\mathbf{W}\mathbf{\Sigma}^{-1} = \mathbf{X}'\mathbf{V}. \tag{3.10}$$

Eq. (3.10) is the matrix $x'_{i\cdot}v_{\cdot j}$ of similarities between subjects i in set $\mathcal{R}$ and ancestry j. In this way, we have estimated genotype principal components for all individuals (i.e., those in $\mathcal{U}$ and $\mathcal{R}$) without the problem confounding due to cryptic relatedness.

3.1.3 PC-Relate

Once we have ancestry-corrected principal components for all individuals, we can obtain “corrected” expected genotypes $\mathbb{E}(X_{ik}) = 2p_{ik}$ of SNPs k for populations of admixed individuals i using a method known as PC-Relate [3]. Because in populations with both shared genetic ancestry and cryptic relatedness, we can use the PC estimates v_{kj} of the allele frequencies in each population j to express the scaled genotypes $x_{ij} = \frac{g_{ik}-2p_k}{2p_k(1-p_k)}$ as

$$\mathbb{E}\left(x_{ik} \mid u_{ij}\right) = u_{ij}\lambda_{jj}v_{kj}, \tag{3.11}$$

or in terms of the measured genotypes:

$$\mathbb{E}\left(g_{ik} \mid u_{ij}\right) = 2p_k + u_{ij}\lambda_{jj}v_{kj} \cdot 2p_k\left(1-p_k\right), \tag{3.12}$$

in which p_k is the allele frequency estimated from the entire sample. Eq. (3.12) says that if we make a graph of each individual's genotype $g_{ik} = 0, 1, 2$ at SNP k vs. the (scaled) of the amount the individual's population-j genetic ancestry, the slope $\lambda_{jj}v_{kj}2p_k\left(1-p_k\right)$ should be the expected allele frequency in population j. In practice, the slopes are obtained from linear regression of the observed genotypes on genetic PCs, and the updated expected allele frequencies $\mathbb{E}\left(g_{ik} \mid u_{ij}\right) = 2p_{ik}$ of SNPs k for individuals i are found by the corresponding value on the best-fit hyperplane, given individual i's genetic ancestry u_{ij}. We can then obtain a new genetic relationship matrix:

$$2\hat{\varphi}_{ij} = \frac{\sum_k\left(g_{ik}-2p_{ik}\right)\left(g_{jk}-2p_{jk}\right)}{2\sqrt{p_{ik}\left(1-p_{ik}\right)p_{jk}\left(1-p_{jk}\right)}}. \tag{3.13}$$

This new GRM will be used in Week 3 when we attempt to fit a linear mixed model for the effect of genotype on disease risk; for now we will use it to make an updated table of the relationship between individuals.

3.2 Kinship: practice

3.2.1 Importing data

Now we'll try kinship analysis for ourselves using a simulated GWAS dataset. This dataset was created using the program bioGWAS [4], which uses other programs to simulate haplotypes [5] and phenotypes [6] based on a set of input variants previously associated with breast cancer [7] and made available by the MRC Interactive Epidemiology Unit's Open GWAS Project [8]. This particular vcf file contains 1000 chromosome 10 genotypes at 345,130 variants (304,770 of which are SNPs) generated from 263 1KGP females of European origin. Download and unzip the file:

```
gunzip EUR_BCa.vcf.gz
```

and move it to your desired location. The unzipped file is readable in plain text format, although it is very big.

Next load the following R packages:

```
library(gdsfmt)
library(GWASTools)
library(SNPRelate)
library(GENESIS)
```

Our first task will be to import the vcf file and convert it to gds format. First specify the directory in which you saved your EUR_BCa.vcf. Then define two file names:

```
vcf.fn <- sprintf("%s/pat_filt_sim.vcf",directory) # vcf file name (exists already)
gds.fn <- sprintf("%s/pat_filt_sim.gds",directory) # gds file name (about to be created)
```

Next perform the conversion and importation into R:

```
snpgdsVCF2GDS(vcf.fn,gds.fn) # create gds file from vcf
genofile <- snpgdsOpen(gds.fn,readonly = FALSE) # import gds object
```

The GDS format is an efficient representation of vcf files which does not require that the entire file be loaded into memory at once [9]. By typing

```
genofile
```

you can see all the fields under your gds object. To access one of these nodes, type

```
index.gdsn(genofile,"sample.id") # vector of sample names
```

or

```
index.gdsn(genofile,"genotype",start = c(1,1), count = c(10,10)) # 10 × 10 sample of the SNP genc
```

This second line uses the start and count options to extract a range of data from the 1000 × 304770 matrix og genotypes. If you have a vector of phenotypes (as will will in Week 3), you can add it as a node in your gds:

```
add.gdsn(genofile,"phenotype",val = fam$Phenotype) # add vector of phenotypes from a table
```

3.2.2 PCA and LD-pruning

For now, we can run PCA on the genotype matrix by simply typing

```
pca <- snpgdsPCA(genofile) # principal components analysis
```

We do not even have to specify the genotype node of genofile. Make a plot of the first few PCs as we did in Week 1.

There are 304770 SNPs in this dataset; many are in LD with their neighbors. To find a set of independent SNPs, which are not in LD ($r > 0.10$) inside a sliding window of size ten million bp, we run:

```
set.seed(566) # set random seed to initiate pruning process
snpset <- snpgdsLDpruning(genofile,method = "corr", slide.max.bp = 10e6,ld.threshold = sqrt(0.1),
pruned <- unlist(snpset, use.names = FALSE) # set of independent SNPs
```

3.2.3 KING IBD and kinship calculation

With our set of pruned SNPs, we can run KING to compute the pairwise kinship coefficient ($\varphi_{ij} = \frac{1}{2}r_{ij}$). We will extract the kinship matrix from the created ibd object and rename the rows and columns with the ids of the samples:

```
ibd <- snpgdsIBDKING(genofile,snp.id = pruned) # KING kinship estimation
colnames(ibd$kinship) <- ibd$sample.id
rownames(ibd$kinship) <- ibd$sample.id
```

Print a 10 × 10 sample of the kinship matrix using:

```
ibd$kinship[1:10,1:10]
```

Do the results make sense? Hint: what should an individual's kinship be with itself?

Finally, we will make a plot of kinship vs. the fraction of the genome shared IBS = 0. This will take a minute or two to plot, because the data consist of all pairwise observations.

```
par(mar = c(5.1,5.1,4.1,2.1) ) # left default plus one
plot(ibd$IBS0,ibd$kinship,ylab = "Kinship coeffecient",xlab = "IBS0",main = "KING rela
```

You should notice that related individuals have a low fraction of IBS = 0, whereas unrelated individuals have a larger proportion. The graph should be approximately linear, but with discontinuities as the relationsips become closer.&nsbp; There should not be many close relatives in this dataset.&nsbp; What, for example, is the largest inferred value of φ?

In addition, if you remove the ylim = c(0,1) option, you will see a large cluster of negative kinship coefficients. These values are expected if the (simulated) individuals are from different genetic backgrounds.

Before proceding to the next step, close your gds file:

```
closefn.gds(genofile)
```

Chapter 4

References

1. Manichaikul A, Mychaleckyj JC, Rich SS, Daly K, Sale M, Chen W-M. Robust relationship inference in genome-wide association studies. Bioinformatics. 2010;26:2867–73.

2. Conomos MP, Miller MB, Thornton TA. Robust Inference of Population Structure for Ancestry Prediction and Correction of Stratification in the Presence of Relatedness. Genetic Epidemiology. 2015;39:276–93.

3. Conomos MP, Reiner AP, Weir BS, Thornton TA. Model-free Estimation of Recent Genetic Relatedness. The American Journal of Human Genetics. 2016;98:127–48.

4. Changalidis AI, Alexeev DA, Nasykhova YA, Glotov AS, Barbitoff YA. bioGWAS: A Simple and Flexible Tool for Simulating GWAS Datasets. Biology. 2023;13:10.

5. Su Z, Marchini J, Donnelly P. HAPGEN2: Simulation of multiple disease SNPs. Bioinformatics. 2011;27:2304–5.

6. Meyer HV, Birney E. PhenotypeSimulator: A comprehensive framework for simulating multi-trait, multi-locus genotype to phenotype relationships. Bioinformatics. 2018;34:2951–6.

7. Michailidou K, BOCS, kConFab Investigators, AOCS Group, NBCS, GENICA Network, et al. Genome-wide association analysis of more than 120,000 individuals identifies 15 new susceptibility loci for breast cancer. Nature Genetics. 2015;47:373–80.

8. Elsworth B, Lyon M, Alexander T, Liu Y, Matthews P, Hallett J, et al. The MRC IEU OpenGWAS data infrastructure. 2020.

9. Zheng X, Levine D, Shen J, Gogarten SM, Laurie C, Weir BS. A high-performance computing toolset for relatedness and principal component analysis of SNP data. Bioinformatics. 2012;28:3326–8.