# PCA practical

## Introduction

In this practical we will do principal component analysis (PCA), which is one of the fundamental tools of population genetics for identifying sample clustering and outliers. As explained in the talk by Gavin PCA is way of reducing a multidimensional space of data points to a set of orthogonal principal components that represent "directions" of greatest variance. These may reflect population structure, but might also reflect cryptic relationships, poor QC, or other effects that lead to differences in genotype distribution.

## The Practical

We will be using the software PLINK written by Christopher Chang:

<https://www.cog-genomics.org/>

We will use plink

* to remove closely-related samples
* to compute principal components
* and to compute the SNP weights or *loadings* that tell us how principal components are weighted across the genome.

We'll also use RStudio to inspect and plot results.

Open a terminal window and navigate to the practical directory:

cd /media/ubuntu/data/GEIA/Practicals/06\_PCA

In this practical, commands that should be run in the terminal will be printed in typewriter font, like the one above.

Also, in RStudio set this directory as your current directory (either using the setwd command or by using the menu option Session->Set working directory->Choose Directory). Commands that should be run in RStudio will be written in a greyed out box, like this:

**In RStudio:**

**setwd('/media/ubuntu/data/GEIA/Practicals/06\_PCA')**

### A note on quality control

Before carrying out a genetic analysis, like PCA, it's important to have a good-quality dataset, and this typically means carrying out careful quality control (QC) first. On this course we'll cover QC in the lectures and practicals tomorrow. For this practical we'll use an already-cleaned dataset contained in the file chr19-clean.vcf.gz. You can look at the data in this file by typing

less -S chr19-clean.vcf.gz

in the terminal, and scroll around using the arrow keys. The data consists of genotype calls at different sites. When you've finished, press the 'q' key to quit back to the terminal prompt.

### LD pruning of SNPs

As described in the population genetics lecture this morning, genetic drift and other processes lead to linkage disequilibrium (LD) between SNPs along a chromosome. To ensure the PCs we compute represent genome-wide structure (not local LD) we'll first carry out LD pruning of our SNP set. This removes correlated pairs of SNPs so that the remaining SNPs are roughly independent. (It also makes subsequent computations quicker.) Run the following command to prune the dataset:

plink --vcf chr19-clean.vcf.gz --maf 0.01 --indep-pairwise 50 5 0.2 --out chr19-clean

**Note:** commands like this should be typed out or pasted onto a single line in the terminal window. Then press <Enter> to run the command. If all goes well you'll see a bunch of output on the screen, starting with some information about the version of plink that is running.

The above command tells plink to load the file chr19-clean.vcf.gz and to prune SNPs to leave SNPs with MAF at least 1%, with no pairs remaining with r2>0.2. (The other parameters, here 50 and 5, affect how the computation works in windows across the genome. You can read about the behaviour here: <http://www.cog-genomics.org/plink2/ld>).

**Q. Look at the screen output from the above plink command. How many variants were in the original dataset? How many were removed because their frequency was below 1%? How many variants were removed due to LD pruning? How many variants remain?**

Type ls or use the file manager to view the directory. The command above produced a number of files that all begin with the chr19-clean prefix. For our purposes, the most important one is chr19-clean.prune.in, as this lists the SNPs that remain after pruning. Feel free to look at these files using less or a text editor.

### IBD pruning of samples / identification of close relationships

We want our top PCs to reflect structure across the majority of our GWAS dataset. Although all samples in our dataset are nominally unrelated, a few duplicated or related samples may have slipped in through the sampling process or through sample handling. We'll therefore first identify and remove any close relationships before computing PCs.

To do this, let's use plink to compute the relatedness between samples. We could do this using the relatedness matrix that we will construct to compute PCA. However, here we'll use genome-wide estimates of identity by descent (IBD) instead. To do this we use the following command:

plink --vcf chr19-clean.vcf.gz --genome gz --out chr19-clean --extract chr19-clean.prune.in

The --genome option tells plink to compute genome-wide IBD estimates (as this file is rather large, we ask plink to store it in a gzipped file). We've used the --extract option to tell plink to only use the LD-pruned set of SNPs we computed above. Further details of the options for computing relatedness and file formats are described here: <https://www.cog-genomics.org/plink2/ibd>.

The output file from the command above is in chr19-clean.genome.gz. Let's load that into RStudio and have a look at it.

**In RStudio:**

ibd <- read.table("chr19-clean.genome.gz", hea=T, as.is=T)

View(ibd)

hist( ibd$PI\_HAT, breaks = 100 )

You can also zoom in along the y-axis to see any close relationships:

hist( ibd$PI\_HAT, breaks = 100, ylim = c(0,1000) )

**Q. Which samples have the closest relationship in the dataset? What is the average relationship between samples in this dataset?**

**Q. In the histogram, there's a tail or 'bump' of relationships up to about 0.2. What could this represent?**

To compute PCs, we'll pick one sample from each closely-related pair and exclude it. (For now we'll just pick the second sample; a more refined approach might look at genotyping performance across the genome and exclude the sample with greater missingness). We'll store the results in a file that we can tell plink about in later steps.

**In RStudio:**

exclusions = ibd[ ibd$PI\_HAT > 0.2, c('FID2','IID2')]

write.table( exclusions, file="related\_samples.txt", col.names = F, row.names = F, quote = F )

**Q. How many samples will be excluded?**

*Note:* the same sample might appear twice in the exclusions data frame. This would happen if they are closely related to more than one other sample. Are there any samples like that? You can use unique(exclusions) to get a list of unique samples to exclude.

### Calculation of principal components

We have now carried out LD and MAF pruning of the variants and we have also identified closely related samples. Now let's use plink to compute PCs.

plink --vcf chr19-clean.vcf.gz --extract chr19-clean.prune.in --remove related\_samples.txt --pca var-wts --out chr19-clean

Leave this to run for a minute or so.

**Q. It's always worth inspecting screen output to check things look right. For example, has plink excluded the right number of samples we told it to?**

Towards the end of the output plink will tell us where it has saved the results. These are in the files chr19-clean.eigenvec (which stores the actual PCs),

chr19-clean.eigenvec.var (which stores the SNP weights or *loadings,* reflecting how much each SNP contributes to each PC),

and chr19-clean.eigenval (which says how much of the overall genotypic variance each PC explains).

### Plot the principal components

Let's load the PCs and plot them:

**In RStudio:**

pcs = read.table( "chr19-clean.eigenvec" )

View(pcs)

plot( pcs[,3], pcs[,4], xlab = "PC1", ylab = "PC2" )

**Q. What do you see? Is there any obvious structure in the dataset?**

We might also want to plot more than just the top two PCs. Let's plot all pairs of the top 5 PCs:

**In RStudio:**

colnames(pcs)[3:7] = c( "PC1","PC2","PC3","PC4","PC5" )

pairs( pcs[, 3:7] )

*Hint:* In Rstudio, you can click the 'zoom' button above above the plot to get a bigger version.

Our dataset contains samples with different ethnicities that in our dataset are recorded in the first column of the file (plink calls this a 'family ID' but here we've used it to record sampled ethnicities). Make a list of them using table():

table( pcs[,1] )

Different ethnic groups might be genetically distinct, so they might separate on the PCs. Let's replot colouring points by their ethnicity. We'll do this in three steps

1. Turn ethnicities into integers that we can pass to the col argument of plot().
2. Plot using the colours
3. Add a legend to the plot so we can see what is what.

(The lower.panel and xpd arguments below are used to tweak the appearance of the plot and how the legend is plotted. See ?pairs and ?par if you want more information, but don't worry about these for now.)

**In RStudio:**

pcs[,1] = as.factor(pcs[,1])

pcs$colour = as.integer(pcs[,1])

pairs(pcs[,3:7], col=pcs$colour, pch=20, lower.panel=NULL)

legend(0.1, 0.5, col=1:max(pcs$colour), legend = levels(pcs[,1]), pch=20, xpd=NA )

**Q. Do the principal components reflect ethnicities in this dataset? Which ethnicities are separated by which PCs?**

**Q. Do any samples look especially genetically distinct from the other samples? What might be the reason for this? Is the same set of samples outlying on all the PCs? Use your R skills to identify these samples.**

### SNP weights/loadings

We would usually like to be able to interpret PCs as representing 'genome-wide' structure and ancestry of our samples. It's therefore wise to check the influence that each variant has on the principal components. To do this, let's examine the

chr19-clean.eigenvec.var file that plink produced. We'll load it into R and plot loadings across the chromosome.

**In RStudio:**

loadings = read.table("chr19-clean.eigenvec.var")

View(loadings)

Columns 3-22 of this file represent the loadings on PCs 1-20, respectively. Let's plot loadings for the first 5 PCs. (Here we use mfrow to make a plot with multiple rows. The mar command adjusts the plot margins. Again, see ?par for details, but don't worry about these for now.)

**In RStudio:**

par(mfrow=c(5,1), mar = c( 1, 4, 1, 2 ))

for( i in 1:5 ) {

plot( 1:nrow(loadings), abs(loadings[,i+2]), ylab=paste("PC ", i, sep="" ), ylim = c( 1, 10 ) )

}

**Q. Do any SNPs stand out as having high loadings for particular PCs? Which PCs? Look back at your plot of principal components. Do these PCs pick out particular clusters of samples?**

**Q. Go back and compute PCs without excluding related samples. Then re-load and plot the PCs and loadings. What drives the top PCs now? What do the loadings look like?**

### Plotting samples against a global dataset

A common strategy for identifying samples with unusual ancestry is to plot samples against a global reference dataset, such as that produced by the 1000 Genomes project (1000G). A common way to do this is to compute a PCA of the external samples and project the GWAS dataset onto it (this can be done in plink using the --within and --pca-cluster-names options - see [the plink website](https://www.cog-genomics.org/plink2/strat) for details). For this practical, however, we'll take a simpler approach and compute a PCA of all the data together.

To do this, we've created a merged dataset including our GWAS data with the African (AFR), European (EUR) and East Asian (EAS) samples from the 1000G reference panel. To compute PCs,we sue a similar command as before, changing the dataset and exclusions files:

plink --vcf merged.with.1000G.vcf.gz --extract chr19-clean.prune.in --pca var-wts --out merged.with.1000G

As before this creates files names merged.with.1000G.eigenvec, etc.

Let's load the merged PCs and plot them with coloured ethnicities as before. It's worth being careful with colours here, so we'll plot points with a particular colour scheme and use different shapes to distinguish the GWAS and reference panel samples.

**In RStudio:**

pcs = read.table( "merged.with.1000G.eigenvec" )

colnames(pcs)[3:7] = c("PC1","PC2","PC3","PC4","PC5")

pcs[,1] = factor(pcs[,1], levels = c("CAN","FAN", "JAN","RAN","AFR","EUR","EAS"))

palette = c("red2","green2","blue2","yellow3", "grey40","grey70","purple")

pcs$colour = palette[as.integer(pcs[,1])]

pcs$shape = 4

pcs$shape[pcs[,1] %in% c('AFR','EUR','EAS')] = 20

pairs(pcs[,3:7], col=pcs$colour, pch=pcs$shape, lower.panel=NULL)

legend(0.1, 0.5, col=palette, legend = levels(pcs[,1]), pch = c(rep(4,4),rep(20,3)), xpd=NA )

**Q. Which reference panel group do most of the GWAS dataset samples cluster with? Are there any that don't?**

**Q. Use R to identify the samples that seem to cluster in the wrong place. Do you recognise these samples? Which reference panel group do they cluster near? What do you conclude about these samples?**

**Q. Some of the 1000G samples labelled 'AFR' also cluster nearer to the Europeans than others. Why is this?**

*Hint:* Population codes can be found in the fileresources/1000GP\_Phase3.sample. Look up the codes for a couple of samples, and look at the population definitions on the 1000 Genomes website at <http://www.1000genomes.org/category/frequently-asked-questions/population>.

## Key Terms

**Principal Component Analysis** (PCA) is a multivariate analysis method for reducing a set of data points in a multidimensional space to a set of orthogonal principal components; also known as eigenvectors. Each eigenvector is associated with an eigenvalue. The one with the largest absolute eigenvalue accounts for most of the variation in the dataset. A loading is associated with each data point. In the context of genetics we often refer to these as SNP weights.

PLINK is one option for doing PCA. It can operate on a subset of samples and project the complementary samples onto the calculated PCs.

[https://www.cog-genomics.org/plink2/strat#pca](https://www.cog-genomics.org/plink2/strat" \l "pca)

EIGENSOFT is another tool that has the added advantage of being able to automatically remove outliers along one or multiple dimensions; a feature, which we will not be using. EIGENSOFT also implements a randomized approximation algorithm, which makes it useful for large dataset.

<http://www.hsph.harvard.edu/alkes-price/software/>

<https://github.com/DReichLab/EIG>

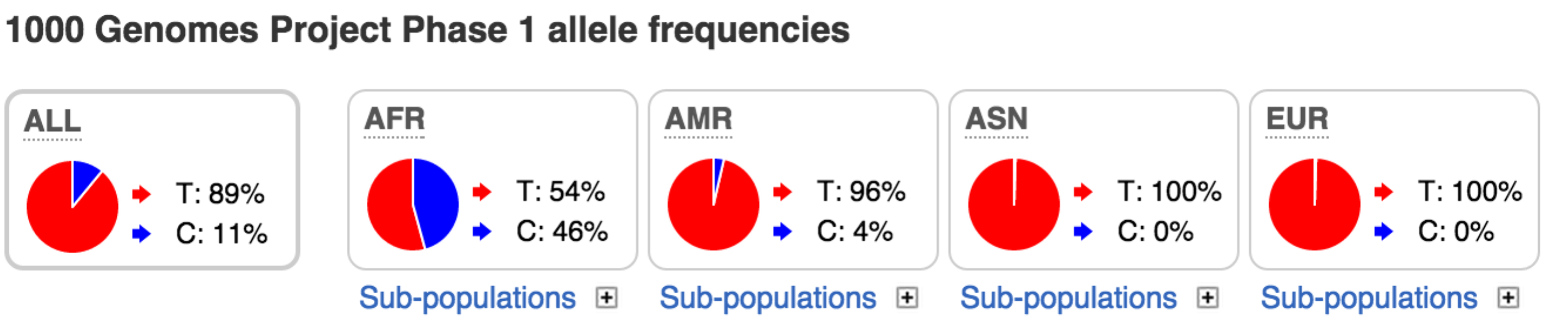
**Allele frequency (AF)** is simply the proportion of one allele type in a population relative to the sum of all alleles in that population. Humans are diploid organisms, so each individual contributes two alleles. The figures below from ensembl.org show allele frequencies for 3 SNPs in the 1000G super populations. PLINK can be used to calculate allele frequencies:

[https://www.cog-genomics.org/plink2/basic\_stats#freq](https://www.cog-genomics.org/plink2/basic_stats" \l "freq)

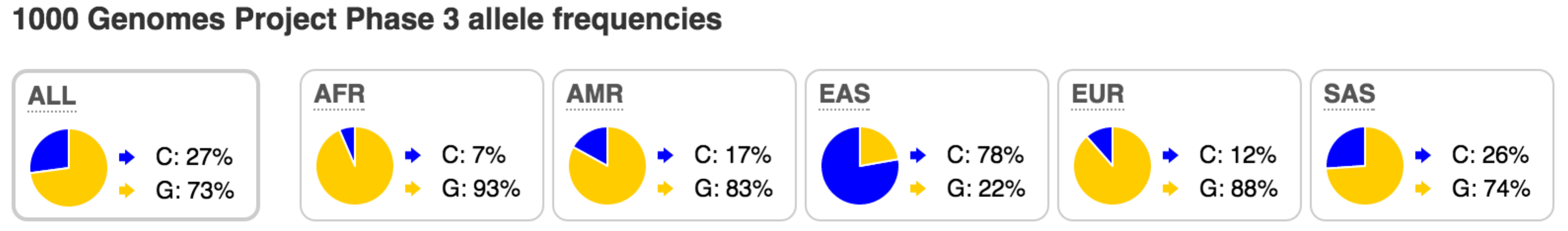
rs6510620



rs60041922

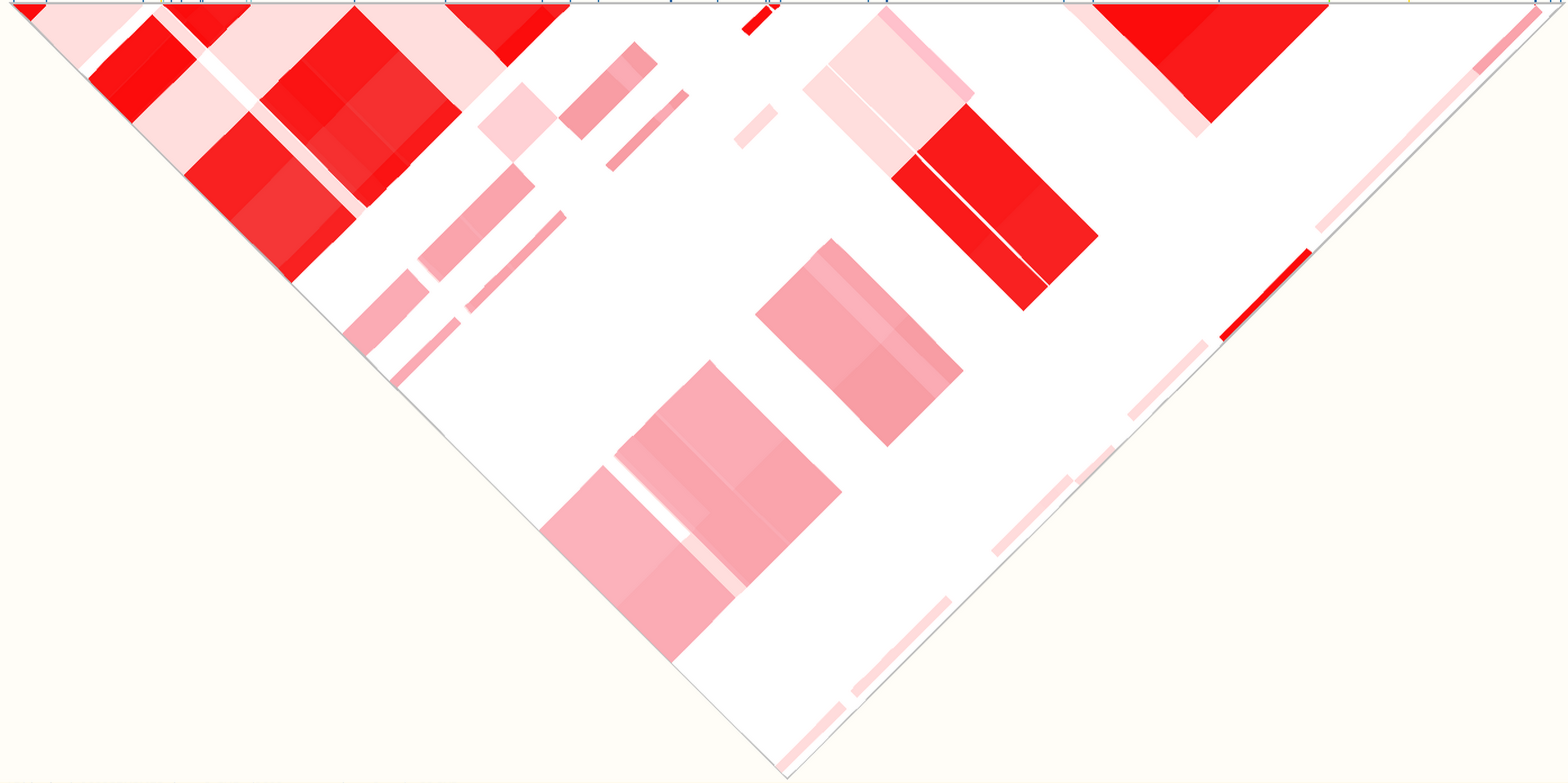


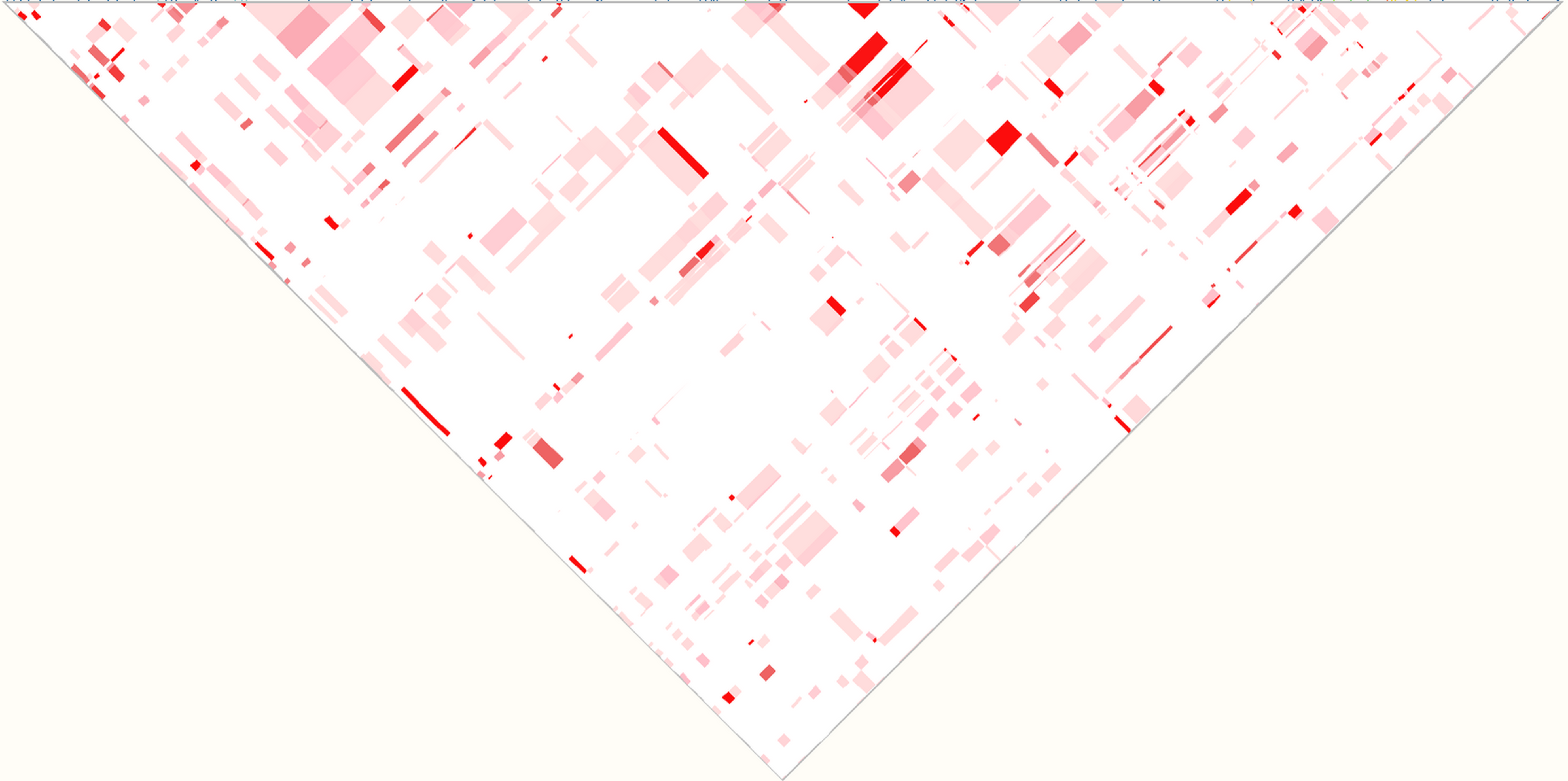
rs806019



**Linkage Disequilibrium** (LD) is a measure of how linked the inheritance of two alleles at different positions are. This is dependent on recombination rates and population structure among other things. The figures below are taken from ensembl.org after searching for rs60041922 and show the correlation coefficient between variants in this region. Dark red symbolises high LD. The top and bottom figure show LD patterns for the 1000G populations CHB (Chinese in Beijing) and YRI (Yoruba in Ibadan), respectively. LD breaks down much faster in the African population, whereas large haplotype blocks are present in the Chinese population. PLINK can be used to calculate LD:

[https://www.cog-genomics.org/plink2/ld#r](https://www.cog-genomics.org/plink2/ld" \l "r)



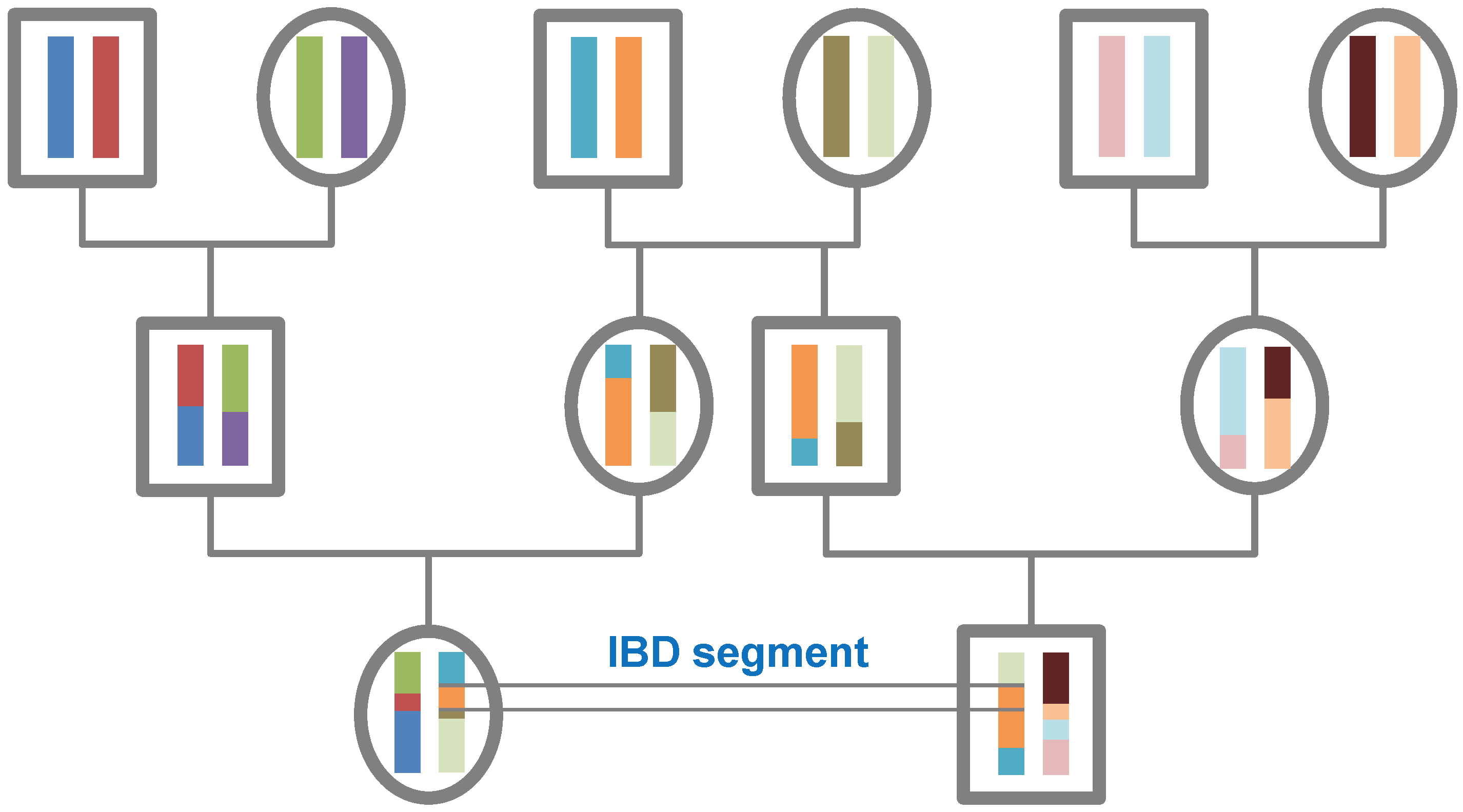


**SNP weight/loading** is a measure of the contribution of a SNP to separate data points along any given PC. SNPs with very different allele frequencies between two datasets often have the greatest SNP weights/loadings. PLINK can be used to calculate SNP weights/loading by adding var-wts after --pca:

[https://www.cog-genomics.org/plink2/strat#pca](https://www.cog-genomics.org/plink2/strat" \l "pca)

**Identity by Descent (IBD)** is the presence of stretches of identical alleles in two individuals, because they inherited that stretch of DNA from the same ancestor. Below an IBD segment is shown in a pedigree involving three sets of grand parents and two 1st cousins. IBD breaks down over generations and in particular in regions with high recombination rates. IBD is a measure of relatedness between individuals. PLINK can do IBD calculations:

<https://www.cog-genomics.org/plink2/ibd>



### PLINK

We will be using the software PLINK. We will be operating on vcf files, but the default PLINK file format is the bed file format. Each .bed file is associated with a .fam and .bim file. The.bed file is a binary file of biallelic genotypes. The .fam file contains Family ID, Individual ID and sex ID among other things. The .bim file contains chromosome ID, variant ID, chromosomal base pair position and allele codes among other things. These files can be modified manually (e.g. family IDs can be changed in the .fam file and variant IDs can be altered in the .bim file), but their original sort order must stay the same, because of their association with the .bed file. You can read more about the file formats here:

[https://www.cog-genomics.org/plink2/formats#fam](https://www.cog-genomics.org/plink2/formats" \l "fam)

[https://www.cog-genomics.org/plink2/formats#bim](https://www.cog-genomics.org/plink2/formats" \l "bim)

[https://www.cog-genomics.org/plink2/formats#bed](https://www.cog-genomics.org/plink2/formats" \l "bed)