1000 Genomes Single Chromosome PCA Example

This example walks through the computation of principal components (PCA) of genomic variant data across one chromosome from 2,504 people from the 1000 genomes project[1]. The example projects all of the variant data for one chromosome into a three-dimensional subspace, and then plots the result. I think the example is popular perhaps because it's very effective at clustering people by ethnicity. It's often used to illustrate "big data" analysis in genomics, even though the data are not particularly big. The point of this example is not to say that PCA on genomic variants is profound, but rather that it's *easy*.

The example uses:

- a very simple C parsing program to efficiently read variant data into an R sparse matrix,
- the irlba package to efficiently compute principal components,
- the threejs package to visualize the result.

NOTE: The example uses commonly available GNU/Linux utilities and shell pipelines (a C compiler, zcat, etc.). It will probably run fine on most Unix systems (including Macs), but not on most Windows systems.
All of these steps, from reading the data in to visualization, only take a few minutes on a decent laptop, and are expressed in just a few lines of R code.

I'd like to thank Dr. David McWilliams for finding bugs and improving these notes.

Reading variant data into an R sparse matrix

This step assumes that you've downloaded and compiled the simple VCF parser and downloaded at least the chromosome 20 and phenotype data files from the 1000 genomes project, for example (from a Mac or Linux shell):

```
# 1000 genomes example variant data file (chromosome 20)
wget ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/release/20130502/ALL.chr20.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz

# 1000 genomes phenotype data file
wget ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/working/20130606_sample_info/20130606_g1k.ped

# Simple but fast parser program (after compilation you'll have a program called a.out)
wget https://raw.githubusercontent.com/bwlewis/1000_genomes_examples/master/parse.c
cc -O2 parse.c
```

We *could* use R alone to read and parse the VCF file, it would just take a while longer.

All the remaining steps in this example run from R. Let’s read the variant data for chromosome 20 into an R sparse matrix. Note that we only care about the variant number and sample (person) number in this exercise and ignore everything else.

```
library(Matrix)
p = pipe("zcat ALL.chr20.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vc
f.gz  | sed /^#/d  | cut  -f '10-' | ./a.out | cut -f '1-2'")
x = read.table(p, colClasses=c("integer","integer"), fill=TRUE, row.names=NULL)

# Convert to a sparse matrix of people (rows) x variant (columns)
chr20 = sparseMatrix(i=x[,2], j=x[,1], x=1.0)

# Inspect the dimensions of this matrix
print(dim(chr20))
# [1]    2504 1812841
```

That was pretty easy! We’ve loaded a sparse matrix with 2,504 rows (people) by 1,812,841 columns (variants). The next step computes the first three principal component vectors using the irlba package and plots a 3d scatterplot using the threejs package. It should run in under a minute even on very modest computers.

```
library(irlba)
cm = colMeans(chr20)
p = irlba(chr20, nv=3, nu=3, tol=0.1, center=cm)

library(threejs)
scatterplot3js(p$u)
```

The data exhibit obvious groups, and those groups correspond to ethnicities. That can be illustrated by loading ancillary data from the 1000 genomes project that identifies the “superpopulation” of each sample.

```
# Read just the header of the chromosome file to obtain the sample identifiers
ids = readLines(pipe("zcat ALL.chr20.phase3_shapeit2_mvncall_integrated_v5a.20130502.
genotypes.vcf.gz  | sed -n /^#CHROM/p | tr '\t' '\n' | tail -n +10"))

# Download and parse the superpopulation data for each sample, order by ids
ped = read.table(url("ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/working/2013
0606_sample_info/20130606_g1k.ped"),sep="\t",header=TRUE,row.names=2)[ids,6,drop=FALS
E]

# Download the subpopulation and superpopulation codes
# WARNING: These links occasionally change. Beware!
pop = read.table("ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/20131219.population
s.tsv",sep="\t",header=TRUE)
pop = pop[1:26,]
super = pop[,3]
names(super) = pop[,2]
super = factor(super)
# The last rows of pop are summary data or non-relevant:

# Map sample sub-populations to super-populations
ped$Superpopulation = super[as.character(ped$Population)]

# Plot with colors corresponding to super populations
N = length(levels(super))
scatterplot3js(p$u, col=rainbow(N)[ped$Superpopulation], size=0.5)
```

```
## Warning: closing unused connection 6 (zcat ALL.chr20.phase3_shapeit2_mvncall_integ
rated_v5a.20130502.genotypes.vcf.gz  | sed -n /^#CHROM/p | tr '    ' '
## ' | tail -n +10)
```

[1] http://www.1000genomes.org/ (http://www.1000genomes.org/)↩