1000 Genomes data, population history

1000 Genomes project provides open data about global human genetic variation. It's such a big project that one can find it with google.

Using web browser, find the variation data for 1000 Genomes project phase 3.

Download the sample list v.3 for the integrated call.

Select samples for "FIN", "CEU" and "YRI". Make text files with the sample identifier only (first field) for each population ("FIN.txt" etc.). Make file "three.txt" that contains sample IDs of all three populations.

How many samples there are for each population?

Extract the variation data for chromosome 20 for these three populations using command:

```
bcftools view -S three.txt ftp://ftp.<URL missing>/<filename missing> -Oz -o
three.chr20.vcf.gz
```

The resulting file is ~85M in size.

Look into the data:

```
bcftools view three.chr20.vcf.gz | less -S
```

See how the AA's have been coded:

```
bcftools view -h three.chr20.vcf.gz | grep AA
```

Select **binary SNPs** and see how it affects the output:

```
bcftools view <fill in options> three.chr20.vcf.gz \
| bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%AA\n' | less
```

That format is not accepted by vcftools. For binary SNPs, we can remove the characters with command `sed 's/|||//'`.

Use bcftools and awk to select a subset of sites; output those in bed format:

```
bcftools view <fill in options> three.chr20.vcf.gz | sed 's/|||//' \
| bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%AA\n' \
| awk '$5!="." && ($5==$3 || $5==$4) {OFS="\t";print $1,$2-1,$2}' \
> sites_aa_match.bed
```

What does this command do?

Select **binary SNPs** for **positions listed in "sites_aa_match.bed" for Finnish samples**; filter the output to remove the extra bars and pipe that to vcftools to compute derived allele frequencies:

```
bcftools view <fill in options> <fill in options> three.chr20.vcf.gz | sed
's/|||//' | vcftools <fill in options> --out FIN_derived
```

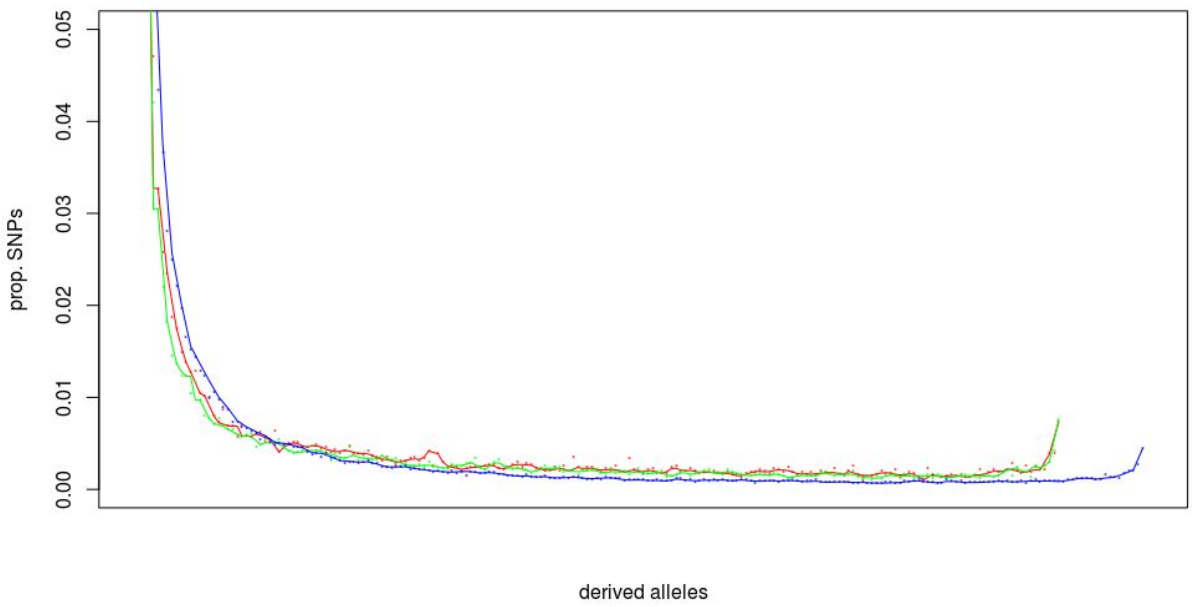
Compute the number of position with different allele counts:

```
awk '{if($4==198){print substr($5,3),substr($6,3)}}' FIN_derived.frq.count \
| sort -n -k2 | uniq -c > FIN_derived.txt
```

Do the same for CEU and YRI.

Make a nice plot on R:

- plot derived allele frequencies (i.e., proportions of total amount) for categories from 1 to (n-1)
- draw these as asterisks using different colors for different populations
- draw a lowess line with "f=0.025"



Reanalysis of human chromosome 20

For this plot, we used all variants in chr 20. In most analysis we would like to study either the signal of functional constraints, or the signal from past demographic structure. Here we want to study the latter.

We make an assumption that variants in intergenic regions are mostly neutral and reflect the evolutionary signal. How do we find "annotation" for "1000 genomes data"?

Find the annotation data and locate the URL for "unfiltered" functional annotation for chromosome 20.

Have a look on the file contents:

```
bcftools view http://ftp.<url>/<file path>/ALL.chr20.phase3_<filename>.vcf.gz \
| less -S
```

Add `bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%GERP\t%CSQ\n'` into the command above.

GERP score is computed with this and best explained here. Zero means neutral expectation, positive values conservation and negative values faster than expected.

Let's select the set of SNPs that (i) intergenic and (ii) have GERP score between 0.5 and -1.

```
bcftools view http://ftp.<url>/<file path>/ALL.chr20.phase3_<filename>.vcf.gz \
-i 'CSQ ~ "intergenic_variant" && GERP < 0.5 && GERP > -1' \
-Oz -o chrom20_intergenic.vcf.gz
```

For simplicity we edit the input vcf file and only keep binary SNPs for which the AA matches either REF or ALT. Using the file we created above:

```
bcftools view -m2 -M2 -v snps -T sites_aa_match.bed three.chr20.vcf.gz \
| sed 's/|||//' | bcftools view -Oz -o three.chr20.binary_snps.vcf.gz
```

and then using the annotation created above:

```
bcftools view -T chrom20_intergenic.vcf.gz three.chr20.binary_snps.vcf.gz -Oz -o
three.chr20.binary_snps.intergenic.vcf.gz
```

(Note that we used -T, not -R. What's the difference between them? You can test it with a small sample if not otherwise clear.)

Now we reduce our sample sizes:

```
head -n 30 FIN.txt > FIN30.txt
```

and similarly for CEU and YRI. Then we compute the derived allele frequencies for each population:

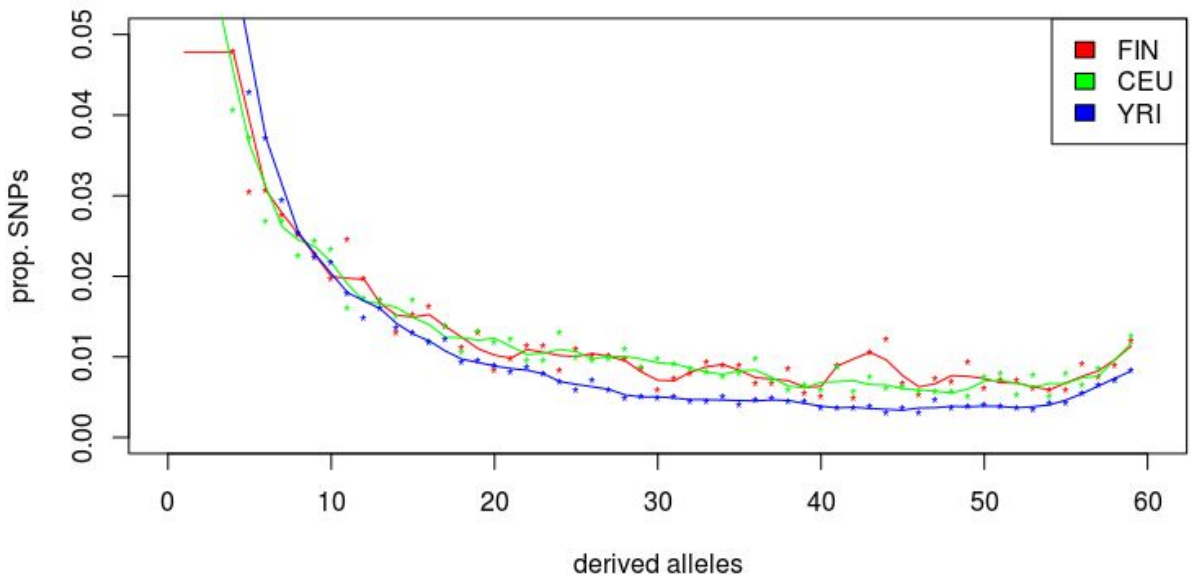
```
bcftools view <insert something> three.chr20.binary_snps.intergenic.vcf.gz |
vcftools <insert something> --out FIN30_derived
```

and collect the numbers of each category:

```
awk '{if($4==60){print substr($5,3),substr($6,3)}}' FIN30_derived.frq.count \
| sort -n -k2 | uniq -c > FIN30_derived.txt
```

Those can now be plotted with R:

```
fin = read.table("FIN30_derived.txt",header=F)
ceu = read.table("CEU30_derived.txt",header=F)
yri = read.table("YRI30_derived.txt",header=F)

colors=rainbow(3)

png("daf_three.png",800,600)

plot(1:59,fin[2:60,1]/sum(fin[2:60,1]),type="p",xlab="derived alleles",ylab="prop.
SNPs",ylim=c(0,0.05),col=colors[1],pch="*",cex=0.75,xlim=c(0,60))
lines(1:59,lowess(fin[2:60,1]/sum(fin[2:60,1]),f=0.1)$y,col=colors[1])
points(1:59,ceu[2:60,1]/sum(ceu[2:60,1]),type="p",col=colors[2],pch="*",cex=0.75)
lines(1:59,lowess(ceu[2:60,1]/sum(ceu[2:60,1]),f=0.1)$y,col=colors[2])
points(1:59,yri[2:60,1]/sum(yri[2:60,1]),type="p",col=colors[3],pch="*",cex=0.75)
lines(1:59,lowess(yri[2:60,1]/sum(yri[2:60,1]),f=0.1)$y,col=colors[3])
legend("topright",c("FIN","CEU","YRI"),fill=colors)

dev.off()
```



In principle, the line should be smooth and bumps in the plot are created (e.g.) demographic events.

One approach to study the demographic history using DAF is program "Stairway plot" (https://sites.google.com/site/jpopgen/stairway-plot). Documentation for the method is available here.

The Stairway plot analysis consists of two parts: (1) \theta estimation and (2) output summary.

In population genetics, \theta is defined as: $\theta=4N\mu$ where N is the population size and \mu is the mutation rate. If we can estimate \theta and know \mu, we can compute *N* which of interest in demographic analyses.

The input format Stairway plot:

```
Input file format:
Columns are separated by TABs.
First row: the first 5 columns are mandatory
       1st col: population id or simulation task name or user's brief note
       2nd col: number of sequences in the sample (nseq)
       3rd col: length of sequence (L)
       4th col: the smallest size of SNP used for estimation. If all SNPs will be
used, then this value is 1.
       5th col: the largest size of SNP used for estimation. If all SNPs will be
used, then this value is nseq-1.
Second row: nseq-1 columns, each is a count of the SNPs of a given size (i.e. \xi).
All counts need to be provided, even though some of them will not be used for
\theta estimation.
       1st col: count of the SNPs of size 1 (i.e. \xi_1)
       2nd col: count of the SNPs of size 2 (i.e. \xi_2)
       ...
       nseq-1_th col: count of the SNPs of size nseq-1 (i.e. \xi_(nseq-1))
```

For the 30 individual samples, the 2nd column is 60 (two chromosome each!). As the length of the sequence we can use 10,263,596. This is a guess based on the fact that the full chr20 is ~65Mb in size and that ~16.3% of the SNPs passed our filtering. The 4th and 5th column are 1 and 59.

Using that, we can create the input file:

```
echo -e "FIN\t60\t10263596\t1\t59" > FIN30.stw
tail -n +2 FIN30_derived.txt | head -n 59 | cut -c -7 | sed 's/ //g' \
| tr '\n' '\t' >> FIN30.stw
echo >> FIN30.stw
```

and then run the first part of the analysis:

```
Stairway_plot_theta_estimation02 FIN30.stw 1 5000
```

When that is finished, we can create the output:

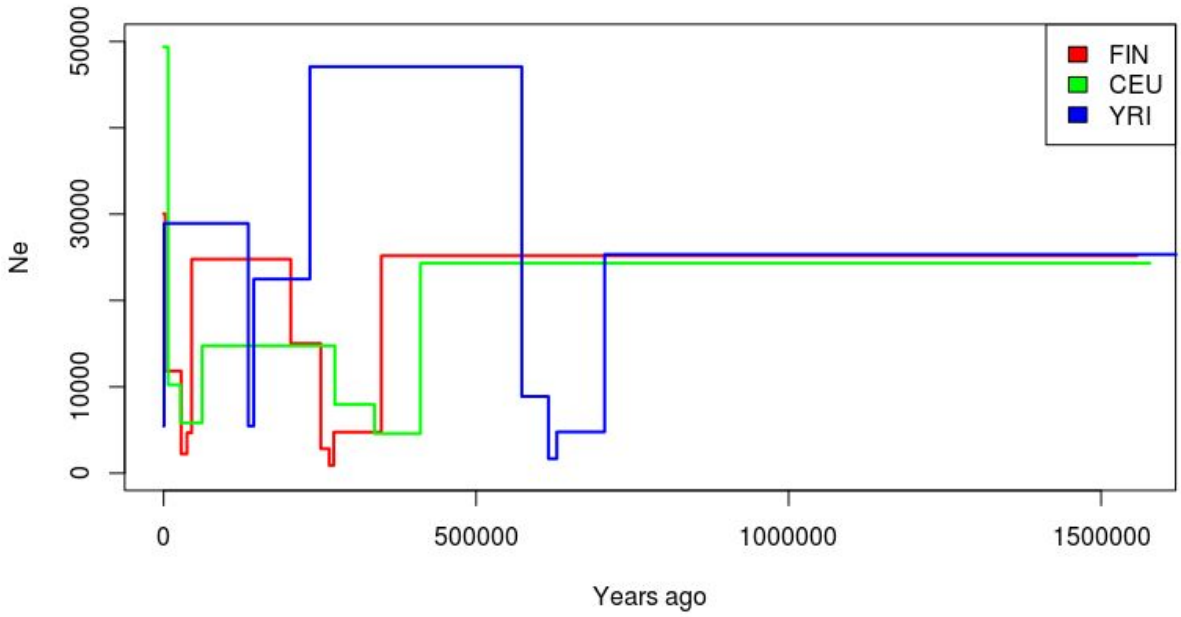
The command line program is easier but it is not on the VM:

```
 mkdir tmp
cp FIN30.stw.addTheta tmp
Stairway_plot_output_summary_commandline tmp 1.2e-8 24 FIN30_stw.out

Stairway_plot_output_summary
```

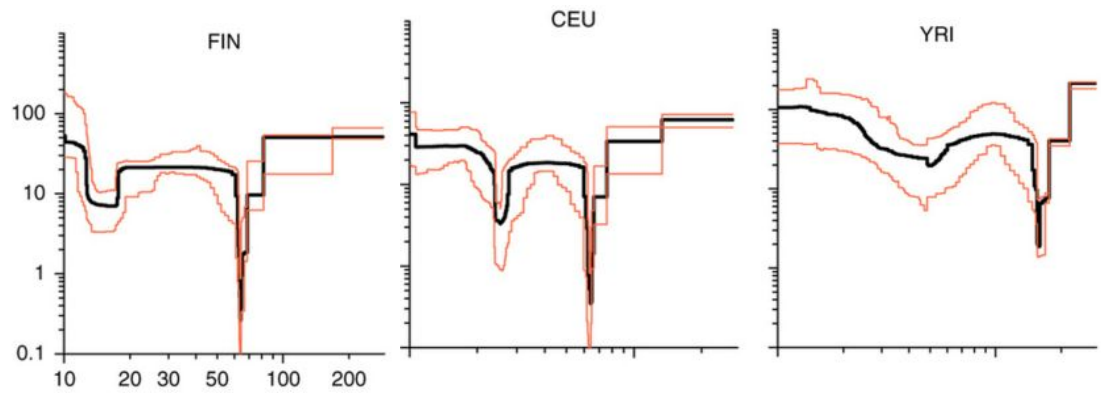
- select "FIN30.stw.addTheta"
- accept mutation rate
- accept generation time
- give output name "FIN30_stw.out"
- do the same for CEU and YRI

When that is done for the three populations, we can plot the past population sizes using R:

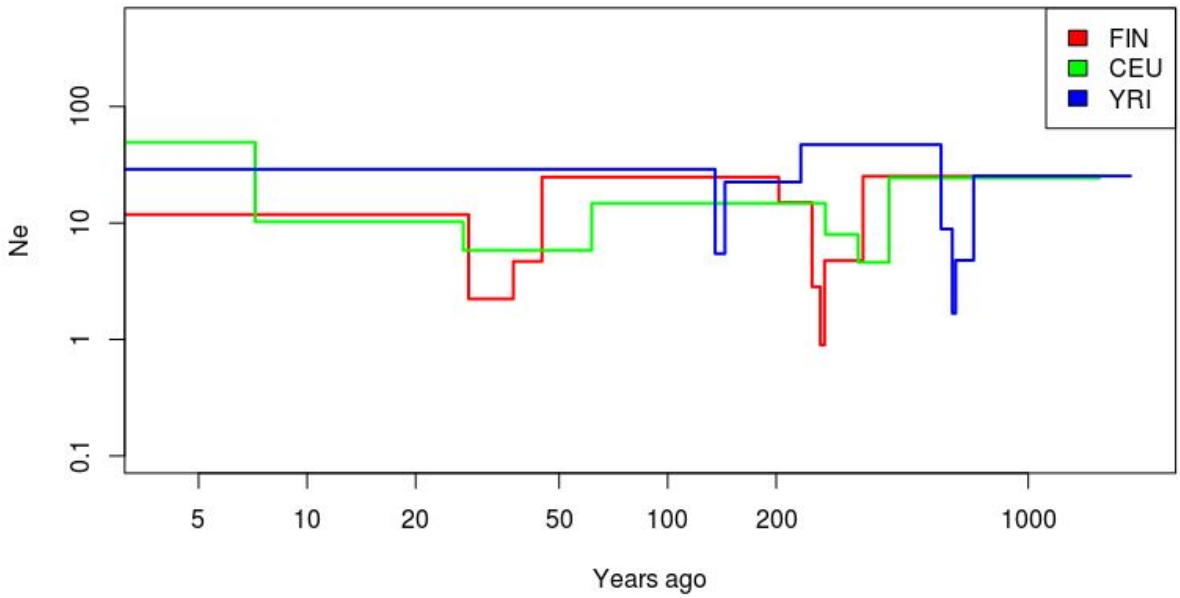
```
fin = read.table("FIN30_stw.out",skip=5,header=T)
ceu = read.table("CEU30_stw.out",skip=5,header=T)
yri = read.table("YRI30_stw.out",skip=5,header=T)

png("stw_three.png",800,600)
colors=rainbow(3)
plot(fin$year,fin$Ne_median,type="l",xlab="Years
ago",ylab="Ne",lwd=2,col=colors[1],ylim=c(0,75000))
lines(ceu$year,ceu$Ne_median,type="l",lwd=2,col=colors[2])
lines(yri$year,yri$Ne_median,type="l",lwd=2,col=colors[3])
legend("topright",c("FIN","CEU","YRI"),fill=colors)
dev.off()
```



One should remember that the analyses are based on very crude filtering and selection of data from one small chromosome only. The results are not consistent with the original results (http://www.nature.com/ng/journal/v47/n5/full/ng.3254.html):



However, if we plot the data in log scale, we see similarities in population bottlenecks but the scaling of x axis in our plot is wrong.



Exam

- probably on the second week of May
- online, probably with lots of time to complete
- computer exercises
- theoretical questions, short essays
- possibly questions based on a scientific article