**Pipeline for preparing two vcf files for painting**

Required files:

1. chr19\_UKB.vcf.gz
2. chr19\_1000G.vcf.gz ---downloaded from <https://bochet.gcc.biostat.washington.edu/beagle/1000_Genomes_phase3_v5a/b37.vcf>
3. refidx.txt --- match file of donor samples and ancestries, downloaded from <https://bochet.gcc.biostat.washington.edu/beagle/1000_Genomes_phase3_v5a/sample_info/integrated_call_samples_v3.20130502.ALL.panel>
4. chr19.map --- downloaded from <https://bochet.gcc.biostat.washington.edu/beagle/genetic_maps>

Required software:

1. bcftools
2. vcftools
3. pbwt
4. Beagle

Steps:

1. Remove SNPs with multi alleles, multi rs names and MAF < 0.5% in both files

bcftools view -Oz -o chr19\_UKB\_filtered\_temp.vcf.gz --exclude 'ID!="." && STRLEN(REF)>1 || N\_ALT>1 || MAF<0.005' chr19\_UKB.vcf.gz

bcftools view -Oz -o chr19\_1000G\_filtered\_temp.vcf.gz --exclude 'ID!="." && STRLEN(REF)>1 || N\_ALT>1 || MAF<0.005' chr19\_1000G.vcf.gz

1. Lift over if necessary

<https://genome.ucsc.edu/cgi-bin/hgLiftOver>

1. Get the intersection of SNPs and update vcf files

bcftools query -f'%POS\n' chr19\_UKB\_filtered\_temp.vcf.gz > UKB\_positions.txt

bcftools query -f'%POS\n' chr19\_1000G\_filtered\_temp.vcf.gz > 1000G\_positions.txt

comm -12 <(sort UKB\_positions.txt) <(sort 1000G\_positions.txt) | sort -n | awk '{print "19\t" $1}' > shared\_positions.txt

bcftools index chr19\_UKB\_filtered\_temp.vcf.gz

bcftools view -R shared\_positions.txt chr19\_UKB\_filtered\_temp.vcf.gz -Oz -o chr19\_UKB\_filtered.vcf.gz

bcftools index chr19\_1000G\_filtered\_temp.vcf.gz

bcftools view -R shared\_positions.txt chr19\_1000G\_filtered\_temp.vcf.gz -Oz -o chr19\_1000G\_filtered.vcf.gz

1. compute the reference allele frequency and draw a scatter plot in R, to check whether they are roughly in a straight line

vcftools --gzvcf chr19\_UKB\_filtered.vcf.gz --freq --out UKB\_freq

vcftools --gzvcf chr19\_1000G\_filtered.vcf.gz --freq --out 1000G\_freq

1. If 4 is fine, then merge these two vcf files

bcftools index -t chr19\_UKB\_filtered.vcf.gz

bcftools index -t chr19\_1000G\_filtered.vcf.gz

bcftools merge -Oz -o merged\_chr19.vcf.gz chr19\_UKB\_filtered.vcf.gz chr19\_1000G\_filtered.vcf.gz

1. Use Beagle to phase the merged vcf file

java -Xmx4g -jar beagle.jar gt=merged\_chr19.vcf.gz map=chr19.map out=merged\_chr19\_phased

1. Split the phased file into 2 files, i.e., separating donor and target samples

bcftools query -l chr19\_UKB\_filtered.vcf.gz > targetname.txt

bcftools query -l chr19\_1000G\_filtered.vcf.gz > donorname.txt

tabix merged\_chr19\_phased.vcf.gz

bcftools view -S targetname.txt -Oz -o chr19\_UKB\_phased.vcf.gz merged\_chr19\_phased.vcf.gz

bcftools view -S donorname.txt -Oz -o chr19\_1000G\_phased.vcf.gz merged\_chr19\_phased.vcf.gz

1. Do imputation if necessary
2. Convert vcf to phase file

pbwt -readVcfGT chr19\_UKB\_phased.vcf.gz -writePhase chr19\_UKB.phase

gzip chr19\_UKB.phase

pbwt -readVcfGT chr19\_1000G\_phased.vcf.gz -writePhase chr19\_1000G.phase

gzip chr19\_1000G.phase

1. Generate genetic map for SNPs

Rscript getmap.R

1. Generate associated files for painting

awk '{print $1, $2}' refidx.txt | awk '!seen[$2]++ {print $2}' | sort > temp\_ancestries.txt

awk '{print $1"\t"NR-1}' temp\_ancestries.txt > ancestry\_to\_number.txt

awk 'NR==FNR {a[$1]=$2; next} {print $1"\t"a[$2]}' ancestry\_to\_number.txt <(awk '{print $1, $2}' refidx.txt) > popnames.txt

1. Do painting: including generating popnames.txt, map.txt, targetname.txt, donor and target phased files (converted using pbwt software).

g++ ../revised\_hashmap.cpp -o test.exe -lz -fopenmp -lpthread

./test.exe