Additional information about the CARTaGENE genetic data

Table of contents

1. GENOTYPING DATA (n=29 337, GSA data only)

1.1 Genotyping data summary

Timeline of genotyping projects at CARTaGENE.

Year	Array Type	Nb*	Array analysis software and details	Comments
2012	Omni 2.5 (HumanOmni2.5-8v1-Multi_A)	937	Genome Studio: 2.0.4 Genome Studio project thresholds (recommended by Illumina): No-call Threshold: 0.15 Clustering Intensity Threshold: 0.20	Returned data generated by Dr P.Awadalla. The following papers should be consulted for details on sample selection: Hodgkinson et al. High-Resolution Genomic Analysis of Human Mitochondrial RNA Sequence Variation. Science. 2014; 344: 413-415. Hussin et al. Recombination affects accumulation of damaging and disease-associated mutations in human populations. Nature Genetics. 2015; 47; 47 (4): 400-404.
2017	Affymetrix Axiom 2.0 UK Biobank gene chip (Axiom_UKB_WCSG-96)	990	Axiom Analysis Suite: 4.0.3.3 Workflow and annotation version: r5	Data generated through a CPTP pilot genotyping project. Caucasians, participants selected according to quantity of health data available
2017	GSAv1 + Multi disease panel (GSAMD-24v1-0_20011747_A1)	5237	Genome Studio: 2.0.4 Genome Studio project thresholds (recommended by Genome Center – Erasmus Medical Center, Netherlands): No-call Threshold: 0.27 Clustering Intensity Threshold: 0.20	This project is an internal genotyping project led by CARTaGENE. Most of the DNA samples came from participants selected according to quantity of health data available; however 2800 participants were selected as part of a research project on restless leg syndrome. Selection criteria for the 1400 cases in this subset were: answered yes to the questions “do you have restless leg syndrome?” or “Have you been diagnosed with restless leg syndrome?”. The 1400 remaining samples from this subset were age/sex-matched controls free of neurological diseases.
2018	GSAv1 (GSA-24v1-0_A1)	726	Genome Studio: 2.0.4 Genome Studio project thresholds (recommended by Illumina): No-call Threshold: 0.15 Clustering Intensity Threshold: 0.20	This project is an internal genotyping project led by CARTaGENE. The DNA samples came from participants selected according to quantity of health data available.
2018	GSAv2 + Multi disease panel (GSAMD-24v2-0_20024620_A)	4179	Genome Studio: 2.0.4 Genome Studio project thresholds (recommended by Illumina): No-call Threshold: 0.15 Clustering Intensity Threshold: 0.20	This project is an internal genotyping project led by CARTaGENE. The DNA samples came from participants selected according to quantity of health data available.

2020	GSAv3 + Multi disease panel (MHI_GSAMD-24v3-0-EA_20034606_C1)	1909	Genome Studio: 2.0.5 Genome Studio project thresholds (recommended by Illumina):	This genotyping project was led by Dr M-P Dubé at the Montreal Heart Institute. Samples previously genotyped using Omni and Axiom where re-genotyped using GSA. This data was QC'ed at CARTaGENE using our regular pipeline.
2021	GSAv2 + Multi disease panel + addon (CaG_addon_v1_20037253_A2)	17286	Genome Studio: 2.0.5 Genome Studio project thresholds (recommended by Illumina):	All remaining samples were selected for this project. The goal was to have genotyping data on all CARTaGENE participants with available blood samples.

*The number of genotypes in the shared data files might be slightly different due to withdrawals from the CARTaGENE study.

Although all genotyping datasets are available, CARTaGENE recommends using the individual GSA datasets (5 datasets) and/or the imputed dataset depending on your research needs.

1.2 Creating the merged dataset (gsa_merged_hg38_20211220) for imputation.

Important note: Variants not present on all GSA array versions have not been removed from the merged dataset. The merged dataset is not suitable for association studies.

The next section details the pipeline used for the quality control of CARTaGENE's genotyping data. The commands, threshold and configurations used are referenced.

Raw data QC

Illumina/Genome Studio genotyping

Missing genders were imported for adequate clustering of SNPs on heterochromosomes. When the genome center that generated the data did not recommend or justify any parameters for the clustering, Illumina recommended parameters were used:

- No-call Threshold = 0.15
- Clustering Intensity Threshold = 0.20

Data was exported in plink format.

Axiom

Axiom Analysis suite configuration:

- Software up to date with latest Axiom UK biobank array annotations
- Best Practice Workflow was used
- Gender File was provided for accurate heterochromosome calls
- Threshold settings: Load settings for Human, then modify:
- Sample QC: all values to 0 (QC will be done with plink)
- SNP QC: cr-cutoff = 0 (QC will be done with plink)

Data was exported in plink format.

Note: Next steps of quality control were performed using Plink v1.90b 6.2 64-bit.

Preliminary QC steps

Remove the control SNPs (chromosome 0) and control samples (used as QC by genomic centers).

```
plink --bfile <INPUT> --not-chr 0 --remove <ARRAY CONTROLS> --make-bed --out <OUTPUT>
```

Rename and set correct gender to samples rescued from identified plate issues (cf. Post QC paragraph).

This is the initial set.

Sample QC

1. Find discordant gender

Compare the sample known gender and the genetic gender imputed with plink:

```
plink --bfile <INPUT> --check-sex --maf 0.02 --make-bed --out <OUTPUT>
```

Samples with discordant genders are removed at the end of the sample quality control.

2. Remove replicates

If some samples were deliberately replicated, the samples with the lower call rates are removed from the set at this step. The call rates are computed with plink:

```
plink --bfile <INPUT> --missing --out <OUTPUT>
```

3. Filter bad quality samples

The SNPs removed during the sample QC are restored at the end of the sample QC to filter them properly without bad quality samples.

- Remove SNPs with a call rate < 95%

```
plink --bfile <INPUT> --geno 0.05 --make-bed --out <OUTPUT>
```

- Remove SNPs failing Hardy-Weinberg test with 10^{-6} threshold

```
plink --bfile <INPUT> --hwe 0.000001 --make-bed --out <OUTPUT>
```

- List samples with a call rate < 95%

```
plink --bfile <INPUT> --mind 0.05 --make-bed --out <OUTPUT>
```

These samples are removed at the end of the sample QC.

- Remove contaminated and duplicated samples

4. SNP pruning

SNPs that are in linkage equilibrium are pruned to reduce the complexity of the pairwise IBD analysis. The IBD analysis excludes uninformative SNPs.

```
plink --bfile <INPUT> --indep-pairwise 50 5 0.5  --out <LD file>
```

5. Pairwise IBD analysis, filter PI_HAT > 0.2

The pairs of samples with a PI_HAT > 0.2 are kept in a list.

```
plink --bfile <INPUT> --exclude <LD FILE>.out --genome --min 0.2 --make-bed --out
<OUTPUT>
```

6. Remove samples similar to at least 50% of the samples

From the IBD 0.2 pair list, samples similar to at least 50% of the samples of the whole set are removed.

These samples are considered contaminated.

They are removed before the next step IBD 0.85.

7. Pairwise IBD analysis, filter PI_HAT > 0.85

Pairs of samples with a PI_HAT > 0.85 are considered duplicates. If the correct sample cannot be identified with absolute certainty, both samples of the pair are eliminated.

plink --bfile <INPUT> --exclude <LD FILE>.out --genome --min 0.85 --make-bed --out <OUTPUT>

Samples flagged as duplicates and contaminated are listed and removed from the initial set.

SNP QC

1. **Remove samples failing the sample QC**

The samples removed for the following step are removed from the initial set:

- Discordant gender
- Sample call rate < 95%
- Contaminants (IBD 0.2, samples paired with 50% of samples)
- Duplicates (IBD 0.85)

This ensures that the SNPs removed later are not influenced by bad quality samples.

2. **Remove SNPs with a call rate < 95%**

```
plink --bfile <INPUT> --geno 0.05 --make-bed --out <OUTPUT>
```

3. **Remove SNPs failing Hardy-Weinberg test with 10^{-6} threshold**

```
plink --bfile <INPUT> --hwe 0.000001 --make-bed --out <OUTPUT>
```

Post QC

The samples that failed the QC are mapped on the plates used for the array analysis. If a pattern is identified, appropriate actions are taken. These actions could be:

- Remove the plate line, column, region or the total plate
- Shift samples: rescue these samples (rename and set correct gender) and redo the QC starting at PRELIMINARY QC STEPS.

1.3 Creating the imputed dataset (imputation_gsa_merged_20220429)

This section details the pipeline used for the creation of CARTaGENE's imputation data. The commands, threshold and configurations used are described in the "Details of commands used".

The quality control and Imputation on TOPMed was perfomed by Ken Sin Lo from the team of Dr Guillaume Lettre.

Data source

The imputation data was prepared from the 5 GSA datasets available as of 2022-04-01 (refer to section 1.1 and 1.2 of this document for QC):

- **GSA_760**: 726 individuals, 626,378 variants
- **GSA_4224**: 4180 individuals, 728,920 variants
- **GSA_5300**: 5239 individuals, 658,297 variants
- **GSA_archi**: 1909 individuals, 688,796 variants
- **GSA_17k**: 17,286 individuals, 645,076 variants

Softwares used

- plink2: plink/2.00-10252019-avx2
- plink: plink/1.9b_6.21-x86_64
- bgzip: tabix/0.2.6
- bcftools: bcftools/1.11
- vcf2gprobs.jar and gprobsmetrics.jar (https://faculty.washington.edu/browning/beagle_utilities/utilities.html)
- R version 4.1.2 (2021-11-01)
- Ruby scripts: align_table.rb, fix_bim.rb, merge_frq.rb, compute_Rsq.rb, filter_mono.rb

Reference population

The reference population used for the imputation is the TOPMed Imputation Reference panel, a diverse reference panel including information from 97,256 deeply sequenced human genomes (https://imputation.biodatacatalyst.nhlbi.nih.gov/#!pages/about).

Data preparation

1. Identify individuals who were genotyped in more than one dataset

There are 3 individuals who are duplicated. Remove all 3 in GSA_760 because there are fewer variants in that dataset.

2. Quality control each of the 5 datasets separately
   a) Run Will Rayner script (https://www.well.ox.ac.uk/~wrayner/strand) to put all alleles on the positive strand.
   b) Remove variant duplicates. Keep the ones with the lowest missingness.
   c) For multi-allelic variants, keep only the one allele with the highest MAF.
   d) Filter for Hardy-Weinberg equilibrium and variant missingness (--hwe 0.000001 midp --geno 0.05).
3. Merge the 5 datasets.
4. Remove these variants from the merged dataset:
   - Monomorphic variants
   - INDELs: variants with D and I as alleles
   - Variants on chr24
   - Variants with a distance greater than 0.075 from the diagonal on the plots comparing allele frequencies between each pair of the 5 datasets
5. Convert the PLINK files to VCF files.
   a) Split the VCF by chromosome for imputation.
   b) Split in 2 batches of ~15,000 individuals selected randomly (because of the limit of 25,000 individuals maximum on the TOPMed server).
6. Imputation on the TOPMed server: https://imputation.biodatacatalyst.nhlbi.nih.gov
7. Merge the 2 batches back together.

   7.a. Compute allele frequencies.

   7.b. Remove monomorphic variants.

   7.c. Compute a merged Rsq (imputation quality score) for the remaining variants.

   7.d. Remove obsolete information from VCFs.

Details of commands used for imputation

1. Identify individuals who were genotyped in more than one dataset.

```
tail -n +2 gsa.4224.final.psam | awk '{ print "4224\t" $0 }' > all_samples.txt
```

9

```
tail -n +2 gsa.5300.final.psam | awk '{ print "5300\t" $0 }' >> all_samples.txt

tail -n +2 gsa.760.final.psam | awk '{ print "760\t" $0 }' >> all_samples.txt

tail -n +2 gsa.archi.final.psam | awk '{ print "1909\t" $0 }' >> all_samples.txt

tail -n +2 gsa.17k.final.hg19.psam | awk '{ print "17286\t" $0 }' >>
all_samples.txt

cut -f 2-4 all_samples.txt | sort | uniq -c -d
```

2. Quality control of each of the 5 datasets separately.

Convert PFILE to BFILE (because of Error: Unrecognized flag ('--update-chr') in PLINK2). The flag '--update-chr' is needed in the next step.

```
plink2 --pfile gsa.4224.final --make-bed --out gsa.4224.final

plink2 --pfile gsa.5300.final --make-bed --out gsa.5300.final

plink2 --pfile gsa.760.final --make-bed --out gsa.760.final

plink2 --pfile gsa.archi.final --make-bed --out gsa.archi.final

plink2 --pfile gsa.17k.final.hg19 --make-bed --out gsa.17k.final
```

2.a. Run Will Rayner script (https://www.well.ox.ac.uk/~wrayner/strand/) to put all alleles on the positive strand.

```
wget https://www.well.ox.ac.uk/~wrayner/strand/update_build.sh

wget https://www.well.ox.ac.uk/~wrayner/strand/GSAMD-24v1-0_20011747_A1-b38-
strand.zip

wget https://www.well.ox.ac.uk/~wrayner/strand/GSA-24v1-0_A1-b38-strand.zip

wget https://www.well.ox.ac.uk/~wrayner/strand/GSAMD-24v2-0_20024620_A1-b38-
strand.zip

wget https://www.well.ox.ac.uk/~wrayner/strand/GSAMD-24v2-0_20024620_B1-b38-
strand.zip

sh strand_Will_Rayner/update_build.sh gsa.4224.final strand_Will_Rayner/GSAMD-24v2-
0_20024620_A1-b38.strand gsa.4224.final.WR_hg38

sh strand_Will_Rayner/update_build.sh gsa.5300.final strand_Will_Rayner/GSAMD-24v1-
0_20011747_A1-b38.strand gsa.5300.final.WR_hg38

sh strand_Will_Rayner/update_build.sh gsa.760.final strand_Will_Rayner/GSA-24v1-
0_A1-b38.strand gsa.760.final.WR_hg38

sh strand_Will_Rayner/update_build.sh gsa.archi.final strand_Will_Rayner/GSAMD-
24v2-0_20024620_B1-b38.strand gsa.archi.final.WR_hg38

sh strand_Will_Rayner/update_build.sh gsa.17k.final strand_Will_Rayner/GSAMD-24v2-
0_20024620_B1-b38.strand gsa.17k.final.WR_hg38
```

2.b. Remove variant duplicates. Keep the ones with the lowest missingness.

```
cut -f 1,4,5,6 gsa.4224.final.WR_hg38.bim | sort -k1,1V -k2,2n -k3,3V -k4,4V | uniq
-d -c | sed 's/\s\s*/\t/g' > gsa.4224.final.WR_hg38.bim.dup

ruby align_table.rb -a gsa.4224.final.WR_hg38.bim -d 1,4,5,6 -A
gsa.4224.final.WR_hg38.bim.dup -D 3,4,5,6 --intersection -o tmp1

ruby align_table.rb -a tmp1 -d 2 -A gsa.4224.final.WR_hg38.lmiss -B 1 -C ' ' -D 2 -
-intersection -o tmp2
```

Sort tmp2 in Excel to put duplicates with lowest missingness first.

```
ruby align_table.rb -a tmp2 -d 1,4,5,6 -A tmp2 -D 1,4,5,6 --intersection2 -o tmp3

uniq tmp3.table2 > tmp4

ruby align_table.rb -a tmp2 -d 2 -A tmp4 -D 2 --difference -o tmp5

cut -f 2 tmp5.table1_diff > gsa.4224.final.WR_hg38.bim.dup_toremove

plink -bfile gsa.4224.final.WR_hg38 --exclude
gsa.4224.final.WR_hg38.bim.dup_toremove --make-bed --out
gsa.4224.final.WR_hg38.wodup
```

Repeat for the other datasets. For GSA_760, also remove the 3 duplicated individuals:

```
plink -bfile gsa.760.final.WR_hg38 --remove overlap_individuals_toremove --exclude
gsa.760.final.WR_hg38.bim.dup_toremove --make-bed --out gsa.760.final.WR_hg38.wodup
```

2.c. For multi-allelic variants, keep only the one allele with the highest MAF.

Compute minor allele frequencies.

```
plink -bfile gsa.4224.final.WR_hg38.wodup --freq --out gsa.4224.final.WR_hg38.wodup

plink -bfile gsa.5300.final.WR_hg38.wodup --freq --out gsa.5300.final.WR_hg38.wodup

plink -bfile gsa.760.final.WR_hg38.wodup --freq --out gsa.760.final.WR_hg38.wodup

plink -bfile gsa.archi.final.WR_hg38.wodup --freq --out
gsa.archi.final.WR_hg38.wodup

plink -bfile gsa.17k.final.WR_hg38.wodup --freq --out gsa.17k.final.WR_hg38.wodup

paste gsa.4224.final.WR_hg38.wodup.bim <(tail -n +2
gsa.4224.final.WR_hg38.wodup.frq | sed 's/\s\s*/\t/g') > tmp.info.4224

paste gsa.5300.final.WR_hg38.wodup.bim <(tail -n +2
gsa.5300.final.WR_hg38.wodup.frq | sed 's/\s\s*/\t/g') > tmp.info.5300

paste gsa.760.final.WR_hg38.wodup.bim <(tail -n +2 gsa.760.final.WR_hg38.wodup.frq
| sed 's/\s\s*/\t/g') > tmp.info.760

paste gsa.archi.final.WR_hg38.wodup.bim <(tail -n +2
gsa.archi.final.WR_hg38.wodup.frq | sed 's/\s\s*/\t/g') > tmp.info.archi

paste gsa.17k.final.WR_hg38.wodup.bim <(tail -n +2 gsa.17k.final.WR_hg38.wodup.frq
| sed 's/\s\s*/\t/g') > tmp.info.17k
```

Create lists of multi-allelic variants to be excluded in the next step.

```
ruby fix_bim.rb

cp gsa.4224.final.WR_hg38.wodup.bim_FIX gsa.4224.final.WR_hg38.wodup.bim

cp gsa.5300.final.WR_hg38.wodup.bim_FIX gsa.5300.final.WR_hg38.wodup.bim

cp gsa.760.final.WR_hg38.wodup.bim_FIX gsa.760.final.WR_hg38.wodup.bim

cp gsa.archi.final.WR_hg38.wodup.bim_FIX gsa.archi.final.WR_hg38.wodup.bim

cp gsa.17k.final.WR_hg38.wodup.bim_FIX gsa.17k.final.WR_hg38.wodup.bim
```

2.d. Filter for Hardy-Weinberg equilibrium and variant missingness (--hwe 0.000001 midp --geno 0.05).

```
plink -bfile gsa.4224.final.WR_hg38.wodup --hwe 0.000001 midp --geno 0.05 --exclude
gsa.4224.final.WR_hg38.wodup.varToRemove --make-bed --out
gsa.4224.final.WR_hg38.wodup.QC

plink -bfile gsa.5300.final.WR_hg38.wodup --hwe 0.000001 midp --geno 0.05 --exclude
gsa.5300.final.WR_hg38.wodup.varToRemove --make-bed --out
gsa.5300.final.WR_hg38.wodup.QC

plink -bfile gsa.760.final.WR_hg38.wodup --hwe 0.000001 midp --geno 0.05 --exclude
gsa.760.final.WR_hg38.wodup.varToRemove --make-bed --out
gsa.760.final.WR_hg38.wodup.QC

plink -bfile gsa.archi.final.WR_hg38.wodup --hwe 0.000001 midp --geno 0.05 --
exclude gsa.archi.final.WR_hg38.wodup.varToRemove --make-bed --out
gsa.archi.final.WR_hg38.wodup.QC

plink -bfile gsa.17k.final.WR_hg38.wodup --hwe 0.000001 midp --geno 0.05 --exclude
gsa.17k.final.WR_hg38.wodup.varToRemove --make-bed --out
gsa.17k.final.WR_hg38.wodup.QC
```

3. Merge the 5 datasets.

```
plink --merge-list datasets_to_merge.txt --make-bed --out gsa_merged_hg38
```

4. Remove these variants from the merged dataset:

- Monomorphic variants
- INDELs: variants with D and I as alleles
- Variants on chr24
- Variants with a distance greater than 0.075 from the diagonal on the plots comparing allele frequencies between each pair of the 5 datasets

Compute allele counts.

```
plink -bfile gsa.4224.final.WR_hg38.wodup.QC --freq counts --out
gsa.4224.final.WR_hg38.wodup.QC.frq

plink -bfile gsa.5300.final.WR_hg38.wodup.QC --freq counts --out
gsa.5300.final.WR_hg38.wodup.QC.frq
```

```
plink -bfile gsa.760.final.WR_hg38.wodup.QC --freq counts --out
gsa.760.final.WR_hg38.wodup.QC.frq

plink -bfile gsa.archi.final.WR_hg38.wodup.QC --freq counts --out
gsa.archi.final.WR_hg38.wodup.QC.frq

plink -bfile gsa.17k.final.WR_hg38.wodup.QC --freq counts --out
gsa.17k.final.WR_hg38.wodup.QC.frq
```

Create one list of all the variants and their frequency counts.

```
ruby merge_frq.rb

head -n 1 table_frqcount.txt > table_frqcount_clean.txt

tail -n +2 table_frqcount.txt | awk '$1 !~ /^24_/ && $3 != "D" && $3 != "I"' >>
table_frqcount_clean.txt
```

Compute distances in R to define the outliers.

```
frqcount <- read.table("table_frqcount_clean.txt", sep="\t", header=T)

frqcount$af_17k <- frqcount$gsa_17k_count1 / (frqcount$gsa_17k_count1 +
frqcount$gsa_17k_count2)

frqcount$af_760 <- frqcount$gsa_760_count1 / (frqcount$gsa_760_count1 +
frqcount$gsa_760_count2)

frqcount$af_4224 <- frqcount$gsa_4224_count1 / (frqcount$gsa_4224_count1 +
frqcount$gsa_4224_count2)

frqcount$af_5300 <- frqcount$gsa_5300_count1 / (frqcount$gsa_5300_count1 +
frqcount$gsa_5300_count2)

frqcount$af_archi <- frqcount$gsa_archi_count1 / (frqcount$gsa_archi_count1 +
frqcount$gsa_archi_count2)

frqcount$d_17k_760 <- abs(frqcount$af_17k - frqcount$af_760) / sqrt(2)

frqcount$d_17k_4224 <- abs(frqcount$af_17k - frqcount$af_4224) / sqrt(2)

frqcount$d_17k_5300 <- abs(frqcount$af_17k - frqcount$af_5300) / sqrt(2)

frqcount$d_17k_archi <- abs(frqcount$af_17k - frqcount$af_archi) / sqrt(2)

frqcount$d_5300_760 <- abs(frqcount$af_5300 - frqcount$af_760) / sqrt(2)

frqcount$d_5300_4224 <- abs(frqcount$af_5300 - frqcount$af_4224) / sqrt(2)

frqcount$d_5300_archi <- abs(frqcount$af_5300 - frqcount$af_archi) / sqrt(2)

frqcount$d_4224_760 <- abs(frqcount$af_4224 - frqcount$af_760) / sqrt(2)

frqcount$d_4224_archi <- abs(frqcount$af_4224 - frqcount$af_archi) / sqrt(2)

frqcount$d_archi_760 <- abs(frqcount$af_archi - frqcount$af_760) / sqrt(2)
```

Identify obvious outliers using the distance from the diagonal.

```
outl_17k_760 <- subset(frqcount, frqcount$d_17k_760 > 0.075)

outl_17k_4224 <- subset(frqcount, frqcount$d_17k_4224 > 0.075)

outl_17k_5300 <- subset(frqcount, frqcount$d_17k_5300 > 0.075)
```

```
outl_17k_archi <- subset(frqcount, frqcount$d_17k_archi > 0.075)

outl_5300_760 <- subset(frqcount, frqcount$d_5300_760 > 0.075)

outl_5300_4224 <- subset(frqcount, frqcount$d_5300_4224 > 0.075)

outl_5300_archi <- subset(frqcount, frqcount$d_5300_archi > 0.075)

outl_4224_760 <- subset(frqcount, frqcount$d_4224_760 > 0.075)

outl_4224_archi <- subset(frqcount, frqcount$d_4224_archi > 0.075)

outl_archi_760 <- subset(frqcount, frqcount$d_archi_760 > 0.075)
```

Visualize the outliers.

```
library(ggplot2)

library(ggpubr)

p1 <- ggplot() + geom_point(data=frqcount, aes(x=af_17k, y=af_760), size=0.5) +
geom_point(data=outl_17k_760, aes(x=af_17k, y=af_760), color="red", size=0.5) +
theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p2 <- ggplot() + geom_point(data=frqcount, aes(x=af_17k, y=af_4224), size=0.5) +
geom_point(data=outl_17k_4224, aes(x=af_17k, y=af_4224), color="red", size=0.5) +
theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p3 <- ggplot() + geom_point(data=frqcount, aes(x=af_17k, y=af_5300), size=0.5) +
geom_point(data=outl_17k_5300, aes(x=af_17k, y=af_5300), color="red", size=0.5) +
theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p4 <- ggplot() + geom_point(data=frqcount, aes(x=af_17k, y=af_archi), size=0.5) +
geom_point(data=outl_17k_archi, aes(x=af_17k, y=af_archi), color="red", size=0.5) +
theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p5 <- ggplot() + geom_point(data=frqcount, aes(x=af_5300, y=af_760), size=0.5) +
geom_point(data=outl_5300_760, aes(x=af_5300, y=af_760), color="red", size=0.5) +
theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p6 <- ggplot() + geom_point(data=frqcount, aes(x=af_5300, y=af_4224), size=0.5) +
geom_point(data=outl_5300_4224, aes(x=af_5300, y=af_4224), color="red", size=0.5) +
theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p7 <- ggplot() + geom_point(data=frqcount, aes(x=af_5300, y=af_archi), size=0.5) +
geom_point(data=outl_5300_archi, aes(x=af_5300, y=af_archi), color="red", size=0.5)
+ theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p8 <- ggplot() + geom_point(data=frqcount, aes(x=af_4224, y=af_760), size=0.5) +
geom_point(data=outl_4224_760, aes(x=af_4224, y=af_760), color="red", size=0.5) +
theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p9 <- ggplot() + geom_point(data=frqcount, aes(x=af_4224, y=af_archi), size=0.5) +
geom_point(data=outl_4224_archi, aes(x=af_4224, y=af_archi), color="red", size=0.5)
+ theme_light() + xlim(c(0,1)) + ylim(c(0,1))

p10 <- ggplot() + geom_point(data=frqcount, aes(x=af_archi, y=af_760), size=0.5) +
geom_point(data=outl_archi_760, aes(x=af_archi, y=af_760), color="red", size=0.5) +
theme_light() + xlim(c(0,1)) + ylim(c(0,1))

ggarrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, labels="AUTO")
```

```
ggsave("frqcount_outliers.png", width=15, height=10, scale=1)
```

Create the list of outliers to be excluded.

```
write.table(outl_17k_760, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t")

write.table(outl_17k_4224, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)

write.table(outl_17k_5300, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)

write.table(outl_17k_archi, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)

write.table(outl_5300_760, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)

write.table(outl_5300_4224, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)

write.table(outl_5300_archi, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)

write.table(outl_4224_760, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)

write.table(outl_4224_archi, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)

write.table(outl_archi_760, file="frqcount_outliers.txt", row.names=F, quote=F,
sep="\t", append=T, col.names=F)
```

Add INDELs (variants with D and I as alleles) and variants on chr24 to be excluded.

```
awk '$5 == "D" || $6 == "D" || $5 == "I" || $6 == "I" || $1 ~ /^24/'
gsa_merged_hg38.bim | cut -f 2 > tmp1

tail -n +2 frqcount_outliers.txt | cut -f 1 >> tmp1

sort tmp1 | uniq > var_to_exclude_from_merged.txt
```

Also remove monomorphic variants.

```
plink -bfile gsa_merged_hg38 --exclude var_to_exclude_from_merged.txt --maf
0.0000001 --make-bed --out gsa_merged_hg38.QC
```

5. Convert the PLINK files to VCF files.

```
plink -bfile gsa_merged_hg38.QC --recode vcf-iid bgz --out
gsa_merged_hg38.QC.no_chr
```

From TOPMed: If your input data is GRCh38/hg38 please ensure chromosomes are encoded with prefix 'chr' (e.g. chr20).

PLINK removes the 'chr' from the chromosome names, add it “manually”.

```
zcat gsa_merged_hg38.QC.no_chr.vcf.gz | head -n 29 > gsa_merged_hg38.QC.vcf
```

```
zcat gsa_merged_hg38.QC.no_chr.vcf.gz | tail -n +30 | awk '{ print "chr" $0 }' >>
gsa_merged_hg38.QC.vcf

bgzip gsa_merged_hg38.QC.vcf

bcftools index gsa_merged_hg38.QC.vcf.gz
```

5.a. Split the VCF by chromosome for imputation.

```
for i in $(seq 1 23); do echo $i; bcftools filter -r chr$i
gsa_merged_hg38.QC.vcf.gz -Oz -o gsa_merged_hg38.QC.chr$i.vcf.gz; done
```

5.b. Split in 2 batches of ~15,000 individuals selected randomly (because of the limit of 25,000 individuals maximum on the TOPMed server).

```
zcat gsa_merged_hg38.QC.vcf.gz | grep -m 1 CHROM | cut -f 10- | sed 's/\t/\n/g' >
all_samples.txt2

shuf all_samples.txt2 > all_samples.txt2.shuf

head -n 15000 all_samples.txt2.shuf > all_samples.txt2.shuf.batch1

tail -n +15001 all_samples.txt2.shuf > all_samples.txt2.shuf.batch2

for i in $(seq 23); do bcftools view -S all_samples.txt2.shuf.batch1
gsa_merged_hg38.QC.chr$i.vcf.gz -Oz -o gsa_merged_hg38.QC.chr$i.b1.vcf.gz; bcftools
view -S all_samples.txt2.shuf.batch2 gsa_merged_hg38.QC.chr$i.vcf.gz -Oz -o
gsa_merged_hg38.QC.chr$i.b2.vcf.gz; done
```

6. Imputation on the TOPMed server: https://imputation.biodatacatalyst.nhlbi.nih.gov

Settings for the imputation:

1. Reference Panel: TOPMed r2
2. Array Build: GRCh38/hg38
3. Rsq Filter: off
4. Phasing: Eagle v2.4 (phased output)
5. Population: vs. TOPMed Panel
6. Mode: Quality Control & Imputation
7. AES 256 encryption: off
8. Generate Meta-imputation file: on

These chunks were excluded by the imputation server:

8. chunk_4_0190000001_0200000000
9. chunk_9_0040000001_0050000000
10. chunk_14_0010000001_0020000000
11. chunk_21_0000000001_0010000000
12. chunk_X.PAR2_0150000001_0160000000

7. Merge the 2 batches back together.

```
bcftools merge -O v -o chr5.merged.vcf batch1/chr5.dose.vcf.gz
batch2/chr5.dose.vcf.gz
```

7.a. Compute allele frequencies.

```
plink --vcf chr5.merged.vcf --freq --out chr5.merged.freq
```

7.b. Remove monomorphic variants.

```
sed 's/\s\s*/\t/g' chr5.merged.freq.frq | cut -f 2- > chr5.merged.freq.frq.tab

awk '$5 > 0' chr5.merged.freq.frq.tab > chr5.merged.freq.frq.tab.noMono

tail -n +2 chr5.merged.freq.frq.tab.noMono | cut -f 2 | sed 's/:/\t/g' | cut -f 1,2
> chr5.merged.freq.frq.tab.noMono.posi

cat chr5.merged.vcf | ruby filter_mono.rb chr5.merged.freq.frq.tab.noMono.posi
chr5.merged.noMono.vcf
```

7.c. Compute a merged Rsq (imputation quality score) for the remaining variants.

```
cat chr5.merged.noMono.vcf | java -jar vcf2gprobs.jar > chr5.merged.noMono.gprobs

cat chr5.merged.noMono.gprobs | java –jar gprobsmetrics.jar >
chr5.merged.noMono.gprobsmetrics
```

The GPROBSMETRICS output contains the following 8 columns:

4. marker identifier
5. minor allele
6. minor allele frequency
7. allelic r-squared
8. dosage r-squared
9. HWE dosage r-squared
10. Accuracy
11. missing score

7.d. Remove obsolete information from VCFs.

```
bcftools annotate -O z -o chr5.merged.clean.noMono.vcf.gz -x
INFO/AF,INFO/MAF,INFO/R2,INFO/ER2 chr5.merged.noMono.vcf

bcftools index chr5.merged.clean.noMono.vcf.gz
```

2. EXOME SEQUENCING DATA (n=198)

2.1 Exome data summary

The CARTaGENE **exome data** has been generated through 2 PI-lead research projects. Bam files and Fastq files are both available.

Year	Platform	Nb	PI	Technical information	Additional information
2012	illumina	96	P. Awadalla	TruSeq Exome Enrichment and TruSeq DNA LT Sample Prep v2 kits, 100-bp paired-end sequencing on the HiSeq 2000, coverage 40x FASTQ files: raw data Bam files: trimmed reads (Galore), aligned (BWA), PCR duplicates removed (Picard), keep properly paired and uniquely mapped (Picard), realigned and recalibration (GATK)	Hodgkinson et al. High-Resolution Genomic Analysis of Human Mitochondrial RNA Sequence Variation. Science. 2014; 344: 413-415. Hussin et al. Recombination affects accumulation of damaging and disease-associated mutations in human populations. Nat Genet
2013	illumina	102	L. Excoffier	sequenced these 102 individuals at very high coverage (mean 89.5X, range 67X-128X) for 106.5 Mb of exomic and UTR regions, Roche NimbleGen SeqCap EZ Exome + UTR Library kit, paired-end (2x100bp) sequenced on an Illumina HiSeq 2500 425 System Raw data: fastq Bam files: trimmed reads (Galore), aligned (BWA)	Peischl et al. Relaxed selection during a recent human expansion. Genetics. 2018; 208(2):763-777. 47(4): 400-404.

2.2 Creating the VCF files

CARTaGENE offers a VCF of SNPs per set of Exome sequencing. Each VCF was produced using a set of pipelines and tools developed at McGill University and Génome Québec Innovation Centre (MUGQIC), called Genpipes. The SNP calling was performed on Beluga (Compute Canada) using dnaseq pipeline from Genpipes version 3.1.5 (link to dnaseq version 3.1.5). Input files are BAM files from each set.

Steps 22 to 29 from MUGQIC dnaseq pipeline were performed.

3. RNA-SEQ DATA (n=911)

The CARTaGENE **RNA-seq data** has been generated through 1 PI-lead research projects:

Date	Platform	Nb	PI	Capture Kit	See these papers for additional information
2012	illumina sequencing	911	Awadalla	TruSeq RNA Sample Prep kit v2, Paired-end RNA sequencing [100 base pairs (bp)], Illumina HiSeq 2000, 3 samples per lane (708), 6 samples per lane (292) Raw data : fastq Bam files: trimmed reads (Galore), aligned (BWA), PCR duplicates removed (Picard), keep properly paired and uniquely mapped (Picard), realigned and recalibration (GATK)	Fave, M. J. et al. Gene-by-environment interactions in urban populations modulate risk phenotypes. Nat. Commun. 9, 827 (2018).

4. WHOLE GENOME SEQUENCING DATA (n=2184)

4.1 Whole genome sequencing data summary

Year	Platform	Nb	PI	Additionall information
2022	Illumina NovaSeq	2184	Internal CARTaGENE led by G.Lettre S.Gravel	**GenoRefQ project.** Selection criteria: Applied to all selected samples: GWAS data available (QC+) Alive and reachable by email via the CaG portal 4 grandparents from the same country (Canada, Morocco, Haiti) Unrelated (within the GénoRef-Q project) Enriched for: RNA-seq data available Participants included in BALSAC Phase A participants (with more data, including nutritional data) Participants of Moroccan origin: 132 Participants of Haitian origin: 163 Participants of Canadian origin: 1889 Sequencing performed at the Center of Expertise and Services (CES) of Genome Quebec. • "Standard" Illumina protocol (PCR-free, paired-end, 2x150bp). • The analysis of the sequencing data was carried out "automatically" with the DRAGEN platform (illumina). • Variant calling was done by considering all the genomes to create a single vcf ("multi-participant") file per chromosome. • gDNA quantified using Quant-iT™ PicoGreen® dsDNA Assay Kit (Life Technologies) and its integrity assessed on agarose gels. • Libraries were generated from 700ng of gDNA using the NxSeq® AmpFREE Low DNA Library Kit Library Preparation Kit (Lucigen). • Dual-indices adaptors were purchased from IDT. • Libraries were quantified using the KAPA Library Quantification Kits - Complete kit (Universal) (Kapa Biosystems). • Average size fragment was determined using a LabChip GXII (PerkinElmer) instrument. • The libraries were normalized and pooled and then denatured in 0.05N NaOH and neutralized using HT1 buffer. • The pool was loaded at 400pM on a Illumina NovaSeq S4 lane using Xp protocol as per the manufacturer's recommendations. • The run was performed for 2x150cycles (paired-end mode). • A phiX library was used as a control and mixed with libraries at 1% level. • Base calling was performed with RTA v3.4.4. Program bcl2fastq2 v2.20 was then used to demultiplex samples and generate fastq reads.

4.2 Creating the dataset

Software used:

bcftools/1.16

plink/1.9b_6.21-x86_64

plink/2.00a3.6

seGMM v1.3.0 (https://github.com/liusihan/seGMM)

ENSEMBL VARIANT EFFECT PREDICTOR v108.1

Filter VCFs created by Illumina DRAGEN on variant genotyping percent (at least 90%) and keep variants with 2 alleles.

```
for i in $(seq 22); do bcftools view -i 'NS_GT/NS > 0.9 & N_ALT < 3' -O v -o
chr$i.vcf
/lustre06/project/6068353/demallia/CARTaGENE/genorefq/GenoRefQ_msVCF/final_pe
r_region.vcf.chr$i.gz; done

bcftools view -i 'NS_GT/NS > 0.9 & N_ALT < 3' -O v -o chrX.vcf
/lustre06/project/6068353/demallia/CARTaGENE/genorefq/GenoRefQ_msVCF/final_pe
r_region.vcf.chrX.gz
```

Create multi-allelic VCFs, filtering again on variant genotyping percent (at least 90%). These multi-allelic variants are not used for quality control.

```
for i in $(seq 22); do bcftools view -i 'NS_GT/NS > 0.9 & N_ALT > 2' -O v -o
chr$i\_multial.vcf
/lustre06/project/6068353/demallia/CARTaGENE/genorefq/GenoRefQ_msVCF/final_pe
r_region.vcf.chr$i.gz; done

bcftools view -i 'NS_GT/NS > 0.9 & N_ALT > 2' -O v -o chrX_multial.vcf
/lustre06/project/6068353/demallia/CARTaGENE/genorefq/GenoRefQ_msVCF/final_pe
r_region.vcf.chrX.gz
```

Create a PLINK version with all the autosomal chromosomes and chromosome X. Remove variants with "NON_REF" as alternate allele.

```
for i in $(seq 22); do plink --vcf chr$i.vcf --double-id --set-
missing-var-ids @:#:\$1:\$2 --make-bed --keep-allele-order --out
chr$i; done

plink --vcf chrX.vcf --double-id --set-missing-var-ids @:#:\$1:\$2 --
make-bed --keep-allele-order --out chrX


grep NON_REF chr*.bim

chr1.bim:1      chr1:241614787:<NON_REF>:A      0       241614787
<NON_REF>       A

chr2.bim:2      chr2:20164721:<NON_REF>:C       0       20164721
<NON_REF>       C

chr3.bim:3      chr3:51694064:<NON_REF>:C       0       51694064
<NON_REF>       C

chr3.bim:3      chr3:75780178:<NON_REF>:G       0       75780178
<NON_REF>       G

chr7.bim:7      chr7:104614650:<NON_REF>:T      0       104614650
<NON_REF>       T

chr7.bim:7      chr7:156266581:<NON_REF>:G      0       156266581
<NON_REF>       G

chr8.bim:8      chr8:28412522:<NON_REF>:C       0       28412522
<NON_REF>       C

chr8.bim:8      chr8:138692288:<NON_REF>:G      0       138692288
<NON_REF>       G

chr8.bim:8      chr8:143182150:<NON_REF>:CTCTGTGTGTGTGTCTGTG  0
143182150       <NON_REF>       CTCTGTGTGTGTGTCTGTG

chr9.bim:9      chr9:14122192:<NON_REF>:G       0       14122192
<NON_REF>       G

chr9.bim:9      chr9:133734372:<NON_REF>:T      0       133734372
<NON_REF>       T

chr15.bim:15    chr15:42409883:<NON_REF>:G      0       42409883
<NON_REF>       G

chr19.bim:19    chr19:3673017:<NON_REF>:G       0       3673017 <NON_REF>
G

chr20.bim:20    chr20:13675298:<NON_REF>:AG     0       13675298
<NON_REF>       AG


plink --merge-list list_for_merge_chrAutX --exclude range
list_var_to_exclude_NON_REF --make-bed --keep-allele-order --out chrAutX
```

Fix 3 sample IDs that contained errors.

```
mv chrAutX.fam chrAutX.fam_ORI

sed
's/Sample_MPS12347215_MPS12347234_B01/Sample_11101257_MPS12347234_B01/g;s/Sam
ple_MPS12347215_MPS12347234_A01/Sample_11139397_MPS12347234_A01/g;s/Sample_MP
S12347215_MPS12347234_C01/Sample_11127964_MPS12347234_C01/g' chrAutX.fam_ORI
> chrAutX.fam
```

Compute variant missingness and individual missingness.

```
plink -bfile chrAutX --missing --out chrAutX
```

Compute Hardy-Weinberg equilibrium. Not used for filtering because of mixed populations in the dataset.

```
plink -bfile chrAutX --hardy midp --out chrAutX
```

Create a pruned version of the dataset.

```
plink -bfile chrAutX --indep-pairwise 500 100 0.3 --out chrAutX_indep

plink -bfile chrAutX --extract chrAutX_indep.prune.in --make-bed --out
chrAutX.pruned
```

Compute heterozygozity on the pruned version.

```
plink -bfile chrAutX.pruned --het --out chrAutX.pruned
```

Remove these samples because missingness > 0.05.

```
awk '$6 > 0.05' chrAutX.imiss

                               FID                                 IID
MISS_PHENO   N_MISS   N_GENO   F_MISS

  Sample_11100002_MPS12346936_A04   Sample_11100002_MPS12346936_A04
Y  6382957 71528595  0.08924

  Sample_11103481_MPS12346936_G04   Sample_11103481_MPS12346936_G04
Y  6597188 71528595  0.09223
```

```
  Sample_11114586_MPS12347234_E01   Sample_11114586_MPS12347234_E01
Y  5452152 71528595  0.07622

  Sample_11119771_MPS12346936_C04   Sample_11119771_MPS12346936_C04
Y  4748779 71528595  0.06639

  Sample_11130581_MPS12346936_F04   Sample_11130581_MPS12346936_F04
Y  4031631 71528595  0.05636

  Sample_11133890_MPS12346936_E04   Sample_11133890_MPS12346936_E04
Y  7818899 71528595   0.1093

  Sample_11140202_MPS12346936_H04   Sample_11140202_MPS12346936_H04
Y  5032529 71528595  0.07036
```

Remove these samples because their heterozygosities are outliers: abs(Fheterozygosity) > 0.4.

```
awk '$6 > 0.4 || $6 < -0.4' chrAutX.pruned.het

                                FID                                 IID
O(HOM)        E(HOM)         N(NM)            F

  Sample_11103295_MPS12347414_E01   Sample_11103295_MPS12347414_E01
36418556    3.657e+07     36881191      -0.4632
```

Create QC'ed version. Remove monomorphic variants.

```
echo '11100002 11100002' > list_sample_to_remove

echo '11103481 11103481' >> list_sample_to_remove

echo '11114586 11114586' >> list_sample_to_remove

echo '11119771 11119771' >> list_sample_to_remove

echo '11130581 11130581' >> list_sample_to_remove

echo '11133890 11133890' >> list_sample_to_remove

echo '11140202 11140202' >> list_sample_to_remove

echo '11103295 11103295' >> list_sample_to_remove

plink -bfile chrAutX --remove list_sample_to_remove --geno 0.05 --maf
0.0000001 --make-bed --out chrAutX_QC
```

Compute genotype concordance between the WGS dataset and the GSA17k dataset. Concordance was similar for the other 4 GSA datasets.

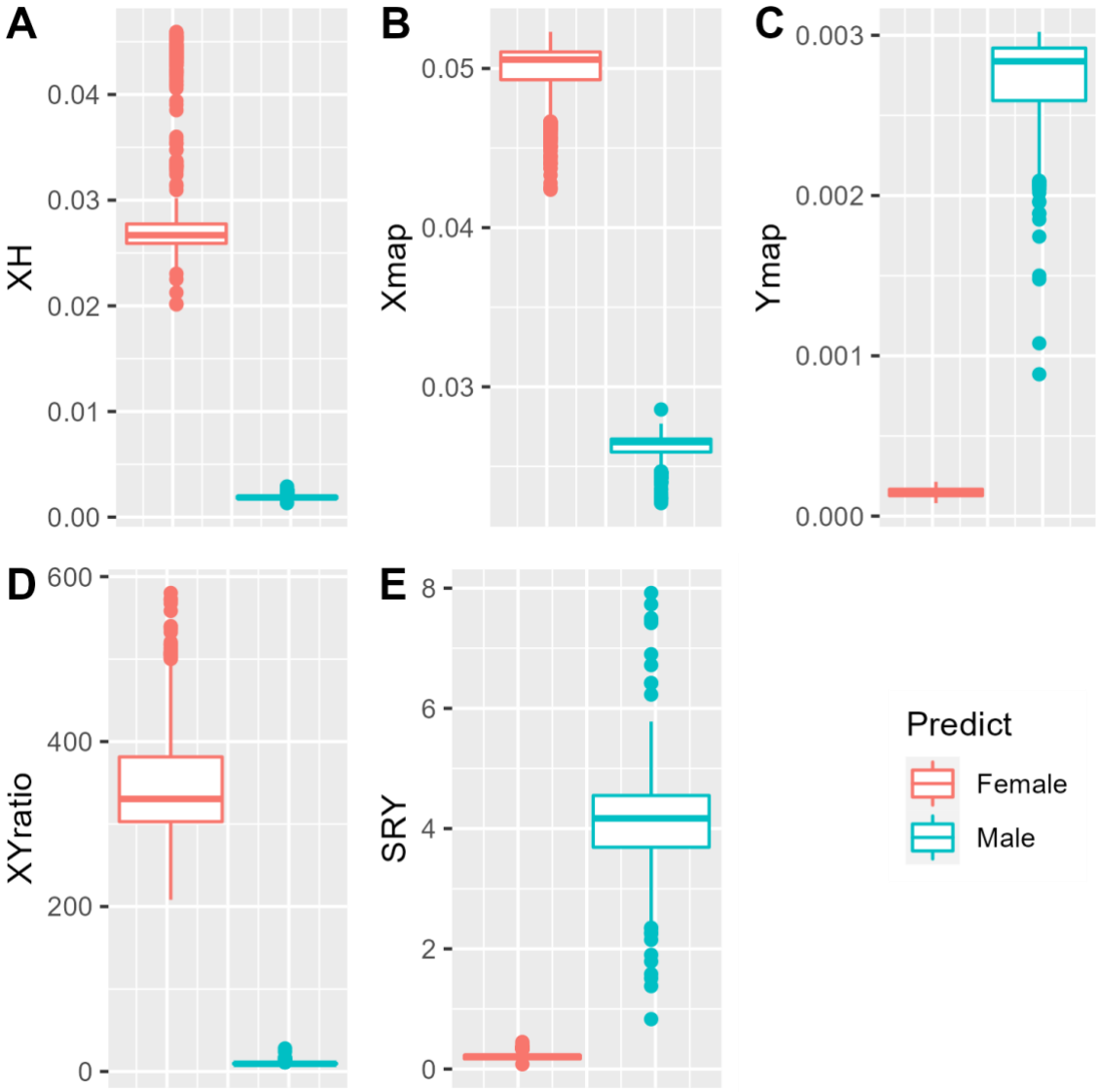
```
mv chrAutX_QC.bim chrAutX_QC.bim_ORI
```

```
awk '{ if ($5 < $6) print $1 "\t" $1 "_" $4 "_" $5 "_" $6 "\t" $3 "\t" $4
"\t" $5 "\t" $6; else print $1 "\t" $1 "_" $4 "_" $6 "_" $5 "\t" $3 "\t" $4
"\t" $5 "\t" $6; }' chrAutX_QC.bim_ORI > chrAutX_QC.bim


plink2 -bfile chrAutX_QC --pgen-diff
gsa.17k.final.WR_hg38.wodup.QC.no_mono.bed
gsa.17k.final.WR_hg38.wodup.QC.no_mono.bim
gsa.17k.final.WR_hg38.wodup.QC.no_mono.fam --out chrAutX_QC.GSA17Kconcordance
```

Compute classification of six sex chromosome karyotypes (XX, XY, XYY, XXY, XXX, and X). No outliers were found.

```
seGMM --vcf chrX.vcf --input list_cram -a CRAM -t WGS -R
resources_broad_hg38_v0_Homo_sapiens_assembly38.fasta -o out_seGMM -n 50
```



Create final version of VCFs. (Codes below are shown for chrX.)

First, we remove the 8 samples identified in the PLINK QC above, variants with missingness > 0.05, variants with NON_REF as alt allele, and monomorphic variants.

We also set the variant IDs, so that it's easy to tell which variants were split from multi-allelic ones to create bi-allelic.

```
bcftools view -i 'NS_GT/NS > 0.95' -S ^list_sample_to_remove_VCF
chrX_multial.vcf | bcftools norm -m -both | bcftools view -i 'ALT[0] !=
"NON_REF"' -c 1 | bcftools annotate --set-id
+'%CHROM\_%POS\_%REF\_%FIRST_ALT\_m' -O v -o chrX_multial_split_QCed.vcf
```

```
bcftools view -i 'NS_GT/NS > 0.95' -S ^list_sample_to_remove_VCF chrX.vcf |
bcftools norm -m -both | bcftools view -i 'ALT[0] != "NON_REF"' -c 1 -O v -o
chrX_bial_step1.vcf

bcftools annotate --set-id +'%CHROM\_%POS\_%REF\_%FIRST_ALT' -O v -o
chrX_bial_QCed.vcf chrX_bial_step1.vcf
```

Merge the bi-allelic part and the multi-allelic split part. Also fix the 3 sample IDs with errors.

```
bcftools concat chrX_bial_QCed.vcf chrX_multial_split_QCed.vcf | bcftools
reheader -s list_sample_to_rename | bcftools sort -T
/scratch/lokensin/tmp_bcftools_sort -O z -o chrX_merged.vcf.gz -m 9G
```

Remove remaining variants with NON_REF as an alternate allele (maybe a bug in bcftools).

```
zcat chrX_merged.vcf.gz | grep -v -w '<NON_REF>' > chrX_FINAL.vcf
```

Compress and index the final VCF.

```
bgzip chrX_FINAL.vcf

tabix -p vcf chrX_FINAL.vcf.gz
```

Compute summary statistics, such as allele frequencies and Hardy-Weinberg equilibrium p-values, per population.

```
for i in $(seq 1 22) X Y; do plink --vcf chr$i\_FINAL.vcf.gz --double-id --
make-bed --keep-allele-order --out chr$i\_FINAL --memory 14800; plink -bfile
chr$i\_FINAL --keep list_sample_FINALVCF_FrenchCanada.ID --freq counts gz --
out chr$i\_FINAL_FrenchCanada --memory 14800; plink -bfile chr$i\_FINAL --
keep list_sample_FINALVCF_Haiti.ID --freq counts gz --out chr$i\_FINAL_Haiti
--memory 14800; plink -bfile chr$i\_FINAL --keep
list_sample_FINALVCF_Morocco.ID --freq counts gz --out chr$i\_FINAL_Morocco -
-memory 14800; done
```

```
for i in $(seq 1 22) X Y; do plink -bfile chr$i\_FINAL --keep
list_sample_FINALVCF_FrenchCanada.ID --hardy midp gz --out
chr$i\_FINAL_FrenchCanada --memory 14800; plink -bfile chr$i\_FINAL --keep
list_sample_FINALVCF_Haiti.ID --hardy midp gz --out chr$i\_FINAL_Haiti --
memory 14800; plink -bfile chr$i\_FINAL --keep
```

```
list_sample_FINALVCF_Morocco.ID --hardy midp gz --out chr$i\_FINAL_Morocco --
memory 14800; done


for i in $(seq 1 22) X Y; do echo $i; zcat chr$i\_FINAL_Morocco.frq.counts.gz
| sed 's/\s\s*/\t/g' | cut -f 2- | head -n 1 | awk '{ printf $0 "\tALT_AF\n"
}' > chr$i\_FINAL_Morocco.frq.counts.af; zcat
chr$i\_FINAL_Morocco.frq.counts.gz | sed 's/\s\s*/\t/g' | cut -f 2- | tail -n
+2 | awk '{ printf $0 "\t%.4g\n", $5 / ($5 + $6) }' >>
chr$i\_FINAL_Morocco.frq.counts.af; gzip chr$i\_FINAL_Morocco.frq.counts.af;
done

for i in $(seq 1 22) X Y; do echo $i; zcat chr$i\_FINAL_Haiti.frq.counts.gz |
sed 's/\s\s*/\t/g' | cut -f 2- | head -n 1 | awk '{ printf $0 "\tALT_AF\n" }'
> chr$i\_FINAL_Haiti.frq.counts.af; zcat chr$i\_FINAL_Haiti.frq.counts.gz |
sed 's/\s\s*/\t/g' | cut -f 2- | tail -n +2 | awk '{ printf $0 "\t%.4g\n", $5
/ ($5 + $6) }' >> chr$i\_FINAL_Haiti.frq.counts.af; gzip
chr$i\_FINAL_Haiti.frq.counts.af; done

for i in $(seq 1 22) X Y; do echo $i; zcat
chr$i\_FINAL_FrenchCanada.frq.counts.gz | sed 's/\s\s*/\t/g' | cut -f 2- |
head -n 1 | awk '{ printf $0 "\tALT_AF\n" }' >
chr$i\_FINAL_FrenchCanada.frq.counts.af; zcat
chr$i\_FINAL_FrenchCanada.frq.counts.gz | sed 's/\s\s*/\t/g' | cut -f 2- |
tail -n +2 | awk '{ printf $0 "\t%.4g\n", $5 / ($5 + $6) }' >>
chr$i\_FINAL_FrenchCanada.frq.counts.af; gzip
chr$i\_FINAL_FrenchCanada.frq.counts.af; done


for i in $(seq 1 22) X Y; do echo $i; zcat chr$i\_FINAL_Morocco.hwe.gz | sed
's/\s\s*/\t/g' | cut -f 2- > chr$i\_FINAL_Morocco.hwe.tab; gzip
chr$i\_FINAL_Morocco.hwe.tab; done

for i in $(seq 1 22) X Y; do echo $i; zcat chr$i\_FINAL_Haiti.hwe.gz | sed
's/\s\s*/\t/g' | cut -f 2- > chr$i\_FINAL_Haiti.hwe.tab; gzip
chr$i\_FINAL_Haiti.hwe.tab; done

for i in $(seq 1 22) X Y; do echo $i; zcat chr$i\_FINAL_FrenchCanada.hwe.gz |
sed 's/\s\s*/\t/g' | cut -f 2- > chr$i\_FINAL_FrenchCanada.hwe.tab; gzip
chr$i\_FINAL_FrenchCanada.hwe.tab; done
```

Annotate the variants using VEP (ENSEMBL VARIANT EFFECT PREDICTOR).

```
for i in $(seq 1 22) X Y; do zcat chr$i\_FINAL.vcf.gz | tail -n +3418 | cut -
f 1-5 > chr$i\_FINAL.vep_in; done


for i in $(seq 1 22) X Y; do singularity exec -B /home/lokensin/projects/def-
glettre/programs/VEP/vep_data:/vep_data,/scratch/lokensin/CARTaGENE/GenoRefQ_
QC_20221206:/wdir --env-file /home/lokensin/projects/def-
glettre/programs/VEP/envfile /home/lokensin/projects/def-
```

```
glettre/programs/VEP/vep.sif /opt/vep/src/ensembl-vep/vep --offline --symbol
--pick --check_existing --af_gnomadg --sift b --polyphen b --force_overwrite
--dir /vep_data --plugin LoF,loftee_path:/vep_data/Plugins -i
/wdir/chr$i\_FINAL.vep_in -o /wdir/chr$i\_FINAL.vep_out; done
```

Annotate the variants using FAVOR; run by the author Hufeng Zhou <hufengzhou@g.harvard.edu>.