Module 1. Identify core samples

The steps in the “Identify core samples” are intended to be executed on each genotype batch after Module 0 has been completed. In this module we identify the core sub-populations (EUR, AFR, ASIAN) in each batch using principal component analysis. For each population, the set of “core samples” is ancestry-matched subset of individuals, with genotypes passing basic QC (i.e. call rated above 95%). Core samples include both parents and offsprings (i.e. not just founders). Of note, EUR population also includes FIN.

# Quality Control (QC) steps

Load PLINK module

Set up the GITHUB environmental variable

GITHUB=/tsd/p697/data/durable/s3-api/github/norment/moba\_qc\_imputation or GITHUB=/cluster/projects/p697/github/norment/moba\_qc\_imputation.

To check the GITHUB variable is set correctly, execute echo $GITHUB command (it should print the full path to the “moba\_qc\_imputation” folder).

Unless otherwise specified all commands are supposed to be run in your working directory (named with your initials) in DATA/DURABLE. Please refer to the File naming and Folder system.docx for the naming conventions.

All plots produced in this module should be copied to /tsd/p697/data/durable/projects/moba\_qc\_imputation/export/Module\_I\_Plots folder when indicated. “Record” in the instructions below refers to recording the numbers in MoBa\_QC\_numbers spreadsheet on Google drive, when indicated. Please contact Elizabeth if you have any doubts/questions about the process.

## Reported pedigree information

Update the fam files to include pedigree information reported through MoBa and the Medical Birth Registry of Norway (MBRN).

### Update the family (FID) and individual (IID) IDs

* + 1. Copy the file to update the FID and IIDs (name: update\_ids.txt, “resources” folder in DATA/DURABLE) to your working folder, this file is a plain text file with 4 columns: the first two columns are the current FIDs and IIDs (SENTRIX IDs) and the third and fourth columns will be the new IDs from update\_ids.txt (In the majority of situations only the FID will change; for duplicates “\_D1” and “\_D2” and for triplicates “\_T1”, “\_T2”, and “\_T3” have been added after the SENTRIX ID to indicate a duplicate/triplicate pair/trio).
    2. Run the following command to update the FID and IIDs:
       1. For HumanCoreExome batches use the command:  
          plink --bfile original-initials-no-black-list --update-ids update\_ids.txt --make-bed --out original-initials-id
       2. For OmniExpress batches whose chromosomal positions did not need updating use the command:  
          plink --bfile original --update-ids update\_ids.txt --make-bed --out original-initials-id
       3. For OmniExpress batches whose chromosomal positions were updated use the command:  
          plink --bfile original-initials-liftover --update-ids update\_ids.txt --make-bed --out original-initials-id
       4. For Global Screening Array batches whose chromosomal positions did not need updating use the command:  
          plink --bfile original-initials-rsids --update-ids update\_ids.txt --make-bed --out original-initials-id
       5. For Global Screening Array batches whose chromosomal positions were updated use the command:  
          plink --bfile original-initials-liftover --update-ids update\_ids.txt --make-bed --out original-initials-id
    3. Record the number of individuals whose IDs were updated (this number can be found in PLINK log file).

### Update paternal (PID) and maternal (MID) IDs

* + 1. Copy the file to update parental IDs (name: update\_parental\_ids.txt, “resources” folder in DATA/DURABLE) to your working folder, this file is a plain text file with 4 columns: the first two columns are the IDs of individuals whose parental IDs we want to update (i.e. children) and the third and fourth columns will be the IDs of their parents (father ID in the third column, mother ID in the fourth column).
    2. Run the following command to update parental IDs:  
       plink --bfile original-initials-id --update-parents update\_parental\_ids.txt --make-bed --out original-initials-parental
    3. Record the number of individuals whose parental IDs were updated (this number can be found in PLINK log file).

### Update sex

* + 1. Copy the file to update sex (name: sex.txt, resources folder) to your working directory; this file is a plain text file with 3 columns, the first two columns are FID and IID, the third column is sex (1=male, 2=female, 0=missing/unknown).
    2. Run the following command to update sex:  
       plink --bfile original-initials-parental --update-sex sex.txt --make-bed --out original-initials-sex
    3. Record the number of individuals whose sex was updated (this number can be found in PLINK log file).
    4. Record the number of individuals with missing sex information.

## Basic QC

### Remove rare variants (MAF>1%)

* + 1. Run the following command to remove rare SNPs:  
       plink --bfile original-initials-sex --maf 0.01 --make-bed --out original-initials-common
    2. Record the number of SNPs removed.

### Call rate

* + 1. Temporarily remove SNPs with call rate below 95%:  
       plink --bfile original-initials-common --geno 0.05 --make-bed --out original-initials-95
    2. Record the number of SNPs removed.
    3. Temporarily remove SNPs and individuals with call rate below 95%:  
       plink --bfile original-initials-95 --geno 0.05 --mind 0.05 --make-bed --out original-initials--call-rates
    4. Record the number of SNPs and individuals removed.

### HWE

* + 1. Temporarily remove SNPs not in HWE (with p<1.00E-03):  
       plink --bfile original-initials--call-rates --hwe 0.001 --make-bed --out original-initials-basic-qc
    2. Record the number of SNPs removed.

## Identify core subsamples

### Prune and remove long stretches of LD

* + 1. Prune the data  
       plink --bfile original-initials-basic-qc --indep-pairwise 3000 1500 0.1 --out original-initials-prune
       1. If needed, repeat the pruning, to remove residual LD, until there are about 100,000 SNPs left (approx 125,000 SNPs and this is fine for running PCA).  
          plink --bfile original-initials-basic-qc --extract original-initials-prune.prune.in --indep-pairwise 3000 1500 0.1 --out original-initials-prune-2
    2. Extract the set of pruned SNPs
       1. If only one round of pruning was run use the following command:  
          plink --bfile original-initials-basic-qc --extract original-initials-prune.prune.in --make-bed --out original-initials-pruned
       2. If you pruned more than once, use the following command:  
          plink --bfile original-initials-basic-qc --extract original-initials-prune-2.prune.in --make-bed --out original-initials-pruned
       3. Record the number of SNPs you have at the end of pruning.
    3. Long LD regions
       1. Copy the file containing the list of long LD regions (Build37/hg19) from the “resources” folder to your working directory (name of the file is “high-ld.txt”). Run the following command lines:
       2. Create high LD set  
          plink --bfile original-initials-pruned --make-set high-ld.txt --write-set --out original-initials-highld
       3. Exclude SNPs in high LD set  
          plink --bfile original-initials-pruned --exclude original-initials-highld.set --make-bed --out original-initials-trimmed

### Identify SNPs overlapping with 1KG

* + 1. Copy the 1KG PLINK bfiles from the “resources” folder in DATA/DURABLE to your working directory (1kg.bed,1kg.bim and 1kg.fam); these files have the strand ambiguous SNPs already removed.
    2. Run the following commands to identify the overlapping SNPs  
       cut -f2 1kg.bim | sort -s > 1kg.bim.sorted  
       cut -f2 original-initials-trimmed.bim | sort -s > original-initials-trimmed.bim.sorted  
       join 1kg.bim.sorted original-initials-trimmed.bim.sorted > original-initials-1kg-snps.txt  
       rm -f 1kg.bim.sorted original-initials-trimmed.bim.sorted
    3. Record the number of SNPs you have common in your batch and in 1KG  
       wc -l original-initials-1kg-snps.txt

### Merge with the 1KG dataset

* + 1. Extract the overlapping SNPs
       1. In your batch:  
          plink --bfile original-initials-trimmed --extract original-initials-1kg-snps.txt --make-bed --out original-initials-1kg-common
       2. In the 1KG dataset:  
          plink --bfile 1kg --extract original-initials-1kg-snps.txt --make-bed --out 1kg-original-initials-common
    2. Merge the bfiles  
       plink --bfile original-initials-1kg-common --bmerge 1kg-original-initials-common --make-bed --out original-initials-1kg-merged
       1. If you have SNPs that have 3+ alleles, flip those alleles in 1kg data and merge again. To flip, run the following command:  
          plink --bfile 1kg-original-initials-common --flip original-initials-1kg-merged-merge.missnp --make-bed --out 1kg-original-initials-flip  
          To merge again, run the following command:  
          plink --bfile original-initials-1kg-common --bmerge 1kg-original-initials-flip --make-bed --out original-initials-1kg-second-merged
       2. If you still have SNPs with 3+ alleles after merging with flipped data, remove those SNPs from both 1KG and MoBa data. To remove the SNPs with 3+ alleles from MoBa and from 1kg, run the following commands:  
          plink --bfile original-initials-1kg-common --exclude original-initials-1kg-second-merged-merge.missnp --make-bed --out original-initials-1kg-clean  
          plink --bfile 1kg-original-initials-flip --exclude original-initials-1kg-second-merged-merge.missnp --make-bed --out 1kg-original-initials-clean  
          To merge again, run the following command:  
          plink --bfile original-initials-1kg-clean --bmerge 1kg-original-initials-clean --make-bed --out original-initials-1kg-clean-merged
    3. Record how many SNPs will be used for PCA:  
       wc -l original-initials-1kg-clean-merged.bim

### PCA using PLINK

* + 1. Copy the populations.txt file from “resources” folder on DATA/DURABLE to your working directory. populations.txt is a text file containing the population, based on which PLINK will calculate the main PCs, in this case it will be just one word “parent”).
    2. Create the original-initials-fam-populations.txt file - a text file with 3 columns: family ID, individual ID and so-called population, in this case it will “parent” for unrelated individuals and “child” for related individuals. To create this file, do the following:  
       awk ‘{if($3==0 && $4==0) print $1,$2,”parent”; else print $1,$2,”child”}’ original-initials-1kg-merged.fam > original-initials-fam-populations.txt
       1. If your data had SNPs with 3+ alleles and you did flip+merge steps, then run the following command:  
          awk ‘{if($3==0 && $4==0) print $1,$2,”parent”; else print $1,$2,”child”}’ original-initials-1kg-second-merged.fam > original-initials-fam-populations.txt
       2. If your data had SNPs with 3+ alleles after flip+merge steps and you did remove+merge steps, then run the following command:  
          awk ‘{if($3==0 && $4==0) print $1,$2,”parent”; else print $1,$2,”child”}’ original-initials-1kg-clean-merged.fam > original-initials-fam-populations.txt
    3. Run the PCA  
       plink --bfile original-initials-1kg-merged --pca --within original-initials-fam-populations.txt --pca-clusters populations.txt --out original-initials-1kg-pca
       1. If your data had SNPs with 3+ alleles and you did flip+merge steps, then run the following command:  
          plink --bfile original-initials-1kg-second-merged --pca --within original-initials-fam-populations.txt --pca-clusters populations.txt --out original-initials-1kg-pca
       2. If your data had SNPs with 3+ alleles after flip+merge steps and you did remove+merge steps, then run the following command:  
          plink --bfile original-initials-1kg-clean-merged --pca --within original-initials-fam-populations.txt --pca-clusters populations.txt --out original-initials-1kg-pca
       3. If your batch is large and you need to run the PCA on Colossus, copy your PLINK input files into Colossus and use the PCA\_POP.job

### Plot PCs

* + 1. The individuals in 1kg.fam file are not ordered according to their population. But in order to assign colors during plotting, we need them to be in order. Thus, we will divide the file containing the PCs into “original” and “1kg” portions, sort the “1kg” portion and combine the two together for plotting. This can be achieved with the following commands:  
       head -n a original-initials-1kg-pca.eigenvec > original-initials-pca-1  
       NB “a” in the “head” command is the number of individuals in the “original” PLINK files that were merged with 1KG (it is the number of individuals in original-initials-1kg-common.fam file).  
       tail -n 1083 original-initials-1kg-pca.eigenvec | sort -k2 > 1kg-initials-pca-1  
       NB 1083 is the number of individuals in 1kg.fam  
       cat original-initials-pca-1 1kg-initials-pca-1 > original-1kg-initials-pca
    2. Plot your batch with the 1KG dataset.
       1. Plot the first 7 PCs in R using the plot-pca-with-1kg.R script. Usage instructions can be found on github (<https://github.com/norment/moba_qc_imputation/tree/master/lib#plot-pca-with-1kgr>). Basic usage: Rscript $GITHUB/lib/plot-pca-with-1kg.R datalabel pcadata legendpos outprefix
          1. Please use the below naming conventions for the Rscript arguments  
             datalabel: Please use the naming convention specified in the /resources/plot-PLINK.txt file  
             pcadata: original-1kg-initials-pca  
             legendpos - possible legend positions are: topleft, topright, bottomleft, bottomright  
             outprefix: original-initials

### Select core subsamples

* + 1. Use either the ellipseselect.py (EUR) or R script (AFR & ASIAN) to select the core subsamples. Usage instructions can be found on github (<https://github.com/norment/moba_qc_imputation/tree/master/lib#select-subsamples-on-pcar> & <https://github.com/norment/moba_qc_imputation/tree/master/software#ellipseselectpy-new-script-to-select-core-subsamples>).
    2. Make PLINK files for each core subsample. The below commands use the file names original-initials-core-subsample-eur.txt, original-initials-core-subsample-afr.txt, and original-initials-core-subsample-asian.txt for selecting the core subsamples. The file names may differ depending on if the R or python script was used.
       1. EUR subsample  
          plink --bfile original-initials-sex --keep original-initials-core-subsample-eur.txt --make-bed --out original-initials-eur
       2. AFR subsample  
          plink --bfile original-initials-sex --keep original-initials-core-subsample-afr.txt --make-bed --out original-initials-afr
       3. ASIAN subsample  
          plink --bfile original-initials-sex --keep original-initials-core-subsample-asian.txt --make-bed --out original-initials-asian
    3. Record the number of individuals you have in each subsample.