Module 3. Merge genotype batches with significant SNP overlap

The steps in Module 3 are intended to be executed on each genotyping batch separately for each core-subsample, before the batches are merged.

# Quality Control (QC) steps

Load PLINK module

Follow the same setup as described in module 1 step-by-step document , e.g. set **GITHUB=/cluster/projects/p697/github/norment/moba\_qc\_imputation** if you work on machine with /cluster access (e.g. p697-submit), or **GITHUB=/tsd/p697/data/durable/s3-api/github/norment/moba\_qc\_imputation** if you work on a machine without /cluster access.

Unless otherwise specified, all commands are supposed to be run in your working directory (named with your initials) in DATA/DURABLE (see folder-structure-moba-2020.pdf)

All plots produced in this module should be copied to /tsd/p697/data/durable/projects/moba\_qc\_imputation/export/Module\_III\_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.

## Create a working directory for running the merge and QC

* 1. Create a working directory for running the merge and QC  
     Example: mkdir /tsd/p697/data/durable/projects/moba\_qc\_imputation/EC/release3
  2. Copy the original-initials-eur-4-batch PLINK bfiles for the relevant batches to your working directory

## Identify SNPs common to all batches

* 1. Use R to identify overlapping SNPs  
     R
     1. Read in bim files, subset the column containing the SNP names, and update the column name.  
        # Below example includes six batches (r1077, r1077, r1108, r1109, r1135, and r1146)  
        R1066 <- read.table(‘original-initials-eur-4-batch.bim’,h=F)  
        R1066 <- data.frame(R1066 [,2])  
        colnames(R1066) <- “SNP”  
        R1077 <- read.table(‘original-initials-eur-4-batch.bim’,h=F)  
        R1077 <- data.frame(R1077 [,2])  
        colnames(R1067) <- “SNP”  
        R1108 <- read.table(‘original-initials-eur-4-batch.bim’,h=F)  
        R1108 <- data.frame(R1108 [,2])  
        colnames(R1108) <- “SNP”  
        R1109 <- read.table(‘original-initials-eur-4-batch.bim’,h=F)  
        R1109 <- data.frame(R1109 [,2])  
        colnames(R1067) <- “SNP”  
        R1135 <- read.table(‘original-initials-eur-4-batch.bim’,h=F)  
        R1135 <- data.frame(R1135 [,2])  
        colnames(R1135) <- “SNP”  
        R1146 <- read.table(‘original-initials-eur-4-batch.bim’,h=F)  
        R1146 <- data.frame(R1146 [,2)  
        colnames(R1067) <- “SNP”
     2. Identify SNPs in all batches  
        # R can only merge 2 data frames at a time. Therefore, perform systematic merging until all batches are merged together.  
        R1066\_R1077 <- data.frame(merge(R1066, R1077, by=”SNP”))  
        rm(R1066, R1077)  
        R1108\_R1109 <- data.frame(merge(R1108, R1109, by=”SNP”))  
        rm(R1108, R1109)  
        R1135\_R1146 <- data.frame(merge(R1135, R1146, by=”SNP”))  
        rm(R1135, R1146)  
        R1066\_R1077\_R1108\_R1109 <- data.frame(merge(R1066\_R1077, R1108\_R1109, by=”SNP”))  
        rm(R1066\_R1077, R1108\_R1109)  
        Release <- data.frame(merge(R1066\_R1077\_R1108\_R1109 , R1135\_R1146, by=”SNP”))  
        rm(R1066\_R1077\_R1108\_R1109, R1135\_R1146)
     3. Create file containing the list of SNP IDs to extract in PLINK bfiles  
        write.table(release, ‘releaseNR\_snps.txt’, quote=F, row.names=F, col.names=F, sep=’\t’) # where NR corresponds to the release number you are analysing   
        q()

## Identify merged families

There are two reasons we need to run this step: 1. some families were genotyped across multiple batches, therefore, if any updates were made to an individual's FID in one batch, we need to ensure that the change is also applied to all other individuals in the initial family; and 2. when KING merges families, a new FID is created starting with the prefix “KING” and a number, starting from with 1 (e.g., KING1, KING2, KING3, …). Therefore, to ensure the family assignment is correct we need to change the FID, otherwise the individuals with the same KING FID would be merged into the same family when we don’t know if they are related or not. For example, if four batches are merged and KING has created a family named “KING1” each batch, all these individuals will be in the KING1 family in the merged dataset, irrespective of whether they are related or not.

* 1. Use R to identify FIDs that have been changed and create update files so the FIDs will be consistent across batches.  
     R  
     library(data.table)  
     library(dplyr)  
     library(tidyr)
     1. Read in fam files, update column names, create a column containing batch information, and subset to FID, IID, and BATCH columns.  
        # Below example includes six batches (r1077, r1077, r1108, r1109, r1135, and r1146)  
        R1066 <- read.table(‘original-initials-eur-4-batch.fam’,h=F)  
        R1066 <- R1066 [,c(1:2)]  
        colnames(R1066) <- c(“FID”,”IID”)  
        R1066$BATCH <- “r1066”

R1077 <- read.table(‘original-initials-eur-4-batch.fam’,h=F)  
R1077 <- R1077 [,c(1:2)]  
colnames(R1077) <- c(“FID”,”IID”)  
R1077$BATCH <- “r1077”  
R1108 <- read.table(‘original-initials-eur-4-batch.fam’,h=F)  
R1108 <- R1108 [,c(1:2)]  
colnames(R1108) <- c(“FID”,”IID”)  
R1108$BATCH <- “r1108”

R1109 <- read.table(‘original-initials-eur-4-batch.fam’,h=F)  
R1109 <- R1109 [,c(1:2)]  
colnames(R1109) <- c(“FID”,”IID”)  
R1109$BATCH <- “r1109”  
R1135 <- read.table(‘original-initials-eur-4-batch.fam’,h=F)  
R1135 <- R1135 [,c(1:2)]  
colnames(R1135) <- c(“FID”,”IID”)  
R1135$BATCH <- “r1135”

R1146 <- read.table(‘original-initials-eur-4-batch.fam’,h=F)  
R1146 <- R1146 [,c(1:2)]  
colnames(R1146) <- c(“FID”,”IID”)  
R1146$BATCH <- “r1146”  
fam <- rbind(R1066, R1077, R1108, R1109, R1135, R1146)

rm(c(R1066, R1077, R1108, R1109, R1135, R1146)

* + 1. Identify FID changes  
       ped <- fread(‘/tsd/p697/data/durable/projects/moba\_qc\_imputation/resources/genotyped\_pedigree.txt’, h=T)  
       ped <- ped[,c(3:4)]  
       match <- merge(ped, fam, by=c(“FID”,“IID”))  
       nonmatch <- fam[!fam$IID %in% match$IID,]  
       colnames(nonmatch) <- c(“KING\_FID”, “IID”, “BATCH”)  
       colnames(ped) <- c(“MoBa\_FID”,“IID”)  
       nonmatch\_ped <- merge(nonmatch, ped, by=“IID”)  
       rm(nonmatch, ped)  
         
       king\_families <- nonmatch\_ped[,c(2,4,3)]  
       king\_families\_u <- distinct(king\_families)  
       king\_families <- king\_families\_u  
       rm(king\_families\_u)  
       freq <- data.frame(table(king\_families$MoBa\_FID))  
       colnames(freq) <- c(“MoBa\_FID”,”freq”)  
       king\_families\_freq <- merge(king\_families, freq, by=”MoBa\_FID”)  
       rm(freq)  
       table(king\_families\_freq$freq)  
       freq1 <- subset(king\_families\_freq, freq==1)  
       freq1 <- freq1[,c(1:3)]  
       freq2 <- subset(king\_families\_freq, freq==2)  
       freq2 <- freq2[,c(1:3)]  
       rm(king\_families\_freq)
    2. Create update ids file with merged FIDs for the families combined by KING  
       The next steps will depend on how many families have been merged by KING for each of the batches. Please use the script /tsd/p697/data/durable/projects/moba\_qc\_imputation/EC/Release3\_freeze/M3/Release3\_king-moba-families.R to identify families updated by king.
       1. This is a very confusing step, ask Elizabeth for help if needed.

## Create bfiles to be merged

* 1. For each batch that is going to be merged run the below plink command  
     plink --bfile original-initials-eur-4-batch --update-ids releaseNR\_king\_updateids.txt --extract releaseNR\_snps.txt --make-bed --out original-initials-eur-releaseNR

## Merge bfiles

* 1. Create a merge list containing the file prefix of all of the batches  
     cat > releaseNR\_merge-list  
     original-initials-eur-releaseNR  
     original-initials-eur-releaseNR  
     original-initials-eur-releaseNR  
     original-initials-eur-releaseNR  
     original-initials-eur-releaseNR  
     original-initials-eur-releaseNR  
     < Control D >
  2. Merge PLINK bfiles  
     plink --merge-list releaseNR\_merge-list --make-bed --out releaseNR-initials-eur
  3. If the merge does not initially work you will need to flip the alleles in at least one batch. It can be a bit finicky trying to identify which batch(es) need to be flipped. Talk to Elizabeth if you need help with this step.