Module 5, 6, and 7: Pre-phasing QC, phasing, and imputation

This document describes steps to perform modules 5, 6, and 7 from the flow-chart, which performs the following steps:

1. Pre-phasing QC
2. Pre-phasing QC report
3. Phasing
4. Imputation
5. Merging imputation chunks across individuals
6. Compute per-SNP imputation INFO scores
7. Merging imputation chunks (dosage in .bgen format)
8. Convert dosages into hard calls format (plink’s format)
9. Filtering out SNPs based on MAF>=0.01 and INFO >=0.8
10. Merging hard calls across chromosomes
11. Imputation report (also saves merged snp stats files to the result folder)

The input to this procedure consists of one “bfile” in plink format (.bed/.bim/.fam), with approximately 400K-800K genetic variants. The output is dosage data (in bgen format, split per chromosome, with 40M genetic variants) and hard calls (in plink “bfile” format, one file concatenated across chromosomes, with between 5M and 10M genetic variants selected by INFO and MAF filters).

# Module 5. Pre-phasing QC

Pre-phasing QC is done by the following script: <https://github.com/norment/moba_qc_imputation/blob/master/jobs/PRE_PHASING_QC.job>

It implements the procedure described here: <https://docs.google.com/document/d/19n7Z6-vv1u0MuUoBZ0F9axLbVYnIKAFiDmgWXKeDqBw/edit>

Steps to run the script as as follows:

1. Copy input bfile to /cluster/projects/p697/projects/moba\_qc\_imputation/OF/<batch>, where <batch> is the name of the imputation batch (i.e. HCE, OMNI, GSA, or similar). This folder will be referred to as <ROOT>.
2. Copy PRE\_PHASING\_QC.job to the <ROOT> folder.
3. Edit PRE\_PHASING\_QC.job by changing “hce-ec-eur-fin-pass-qc” to the actual name of your input bfile
4. Run “sbatch PRE\_PHASING\_QC.job”. This may take approximately 10 minutes.

## Pre-phasing QC report

Pre-phasing QC does not exclude any individuals. Therefore, pre-phasing QC report describes the changes at SNP level - how many SNPs were excluded (and why), how many got a new rs#, flip strand, etc. At this point there are no step-by-step instructions to generate pre-phasing QC report. It’s based on the following script: <https://github.com/norment/moba_qc_imputation/blob/master/report/pre_phasing_qc_report_HCE.ipynb> , but it needs to be adjusted for a given batch. A more recent version of these scripts is available here: <https://github.com/norment/moba_qc_imputation/blob/master/users/of/GSA_may2021/MoBa_pre_phasing_qc_and_imputation_reports.ipynb>

The latest version: /cluster/projects/p697/users/ofrei/jupyter/MoBa\_pre\_phasing\_qc\_and\_imputation\_reports.ipynb

# Module 6. Phasing

Phasing uses HRC reference panel, converted from the original files to a format compatible with shapeit2 using the following script : <https://github.com/norment/moba_qc_imputation/blob/master/jobs/make_hrc_format_conversion_scripts.py> . This is a one-time operation - the result is already stored on TSD, and there is no need to repeat it for a new batch.

Steps to run phasing:

1. Copy <https://github.com/norment/moba_qc_imputation/blob/master/jobs/make_jobs.py> to your <ROOT> folder
2. Run the following command to define chunks of individuals:

python make\_jobs.py ichunks --prefix chr@

Note how many chunks of individuals (‘ichunks’) were generated. This could be confusing: phasing is going to be executed without forming chunks of individuals, but the result of phasing is then split into chunks using the “shapeit2 --convert” command.

1. Run the following command to generate phasing .job files

python make\_jobs.py shapeit2 --prefix chr@ --hours 168 --num-ichunks <N>

1. Submit .job files by running 22 commands like this:

sbatch shapeit2\_chrNN.job

The previous step outputs 22 commands that can be copy-pasted and executed.   
The jobs will take between 1 and 3 days to finish (for a N=30K batch)

for ((NN=1; NN<=22; NN++)); do

sbatch shapeit2\_chr${NN}.job

done

**(!) WARNING (!)** The chunks have individuals re-shuffled to guarantee that families were not split across imputation chunks. A new chr@.ichunk\_merged.fam file was generated at step 2 listing the new order of individuals.

# Module 7. Imputation

Steps to run imputation:

1. Run the following command to generate imputation .job files:

python make\_jobs.py impute4 --prefix chr@ --hours 24 --num-ichunks <N>

1. Submit .job files by running Nx22 commands like this:

sbatch impute4\_chrNN\_ichunkXX.job

Where XX corresponds to chunk number (varies from 1 to N). The previous step outputs Nx22 commands that can be copy-pasted and executed.

**(!) WARNING (!)** If there are too many jobs to execute (roughly Nx924, could exceed the TSD limit of 4500 jobs submitted to the queue per project). The solution is then to submit jobs for a few chromosomes at a time. For example, in order to run imputation jobs for first 5 chromosomes, you may run the following loop:

for ((NN=1; NN<=5; NN++)); do

for ((XX=1; XX<=<N>; XX++)); do

sbatch impute4\_chr${NN}\_ichunk${XX}.job

done

done

1. The jobs will take between 1 and 3 days to finish. We no longer expect jobs to fail due to “out of memory”, but if this happens you would need to re-generate failed jobs with increased memory limit. This can be done with “--missing-only” argument of the “make\_jobs.py impute4”:

python make\_jobs.py impute4 --prefix chr@ --hours 48 --out impute4\_chr@\_4xMEM.job --mem-per-cpu 36000 --missing-only --num-ichunks <N>

1. Check that all batches exist, i.e. by counting the number of bgen files (“ls \*gen.gz | wc -l”), and comparing it to the number of chunks (“python make\_jobs.py impute4 --prefix chr@ --num-ichunks <N> --out temp.job | grep Chunk | wc -l”)

## Merging imputation chunks across individuals

1. Re-generate all .job files by running the following:

python make\_jobs.py impute4 --prefix chr@ --hours 24 --num-ichunks <N>

1. Submit .job files by running 22 commands like this:

sbatch impute4\_chrNN\_merge.job

for ((NN=1; NN<=22; NN++)); do

sbatch impute4\_chr${NN}\_merge.job

done

The previous step outputs 22 commands that can be copy-pasted and executed.

This step creates a set of .bgen files, and calculates imputation INFO scores using qctool. Due few duplicated rs# at the same CHR:BP we apply **rename\_multiallelic\_snps.py** to ensure a unique marker name. This is done before creating .bgen and INFO files - all of them have duplicated IDs replaced with CHR:BP\_A1\_A2 codes. In total this affects about ~10.000 variants in HRC reference, the names for all other variants are kept unchanged.

## Merging imputation chunks (dosage data)

1. Re-generate “Makefile” file by running the following command:

python make\_jobs.py impute4 --prefix chr@ --hours 48 --num-ichunks <N>

1. Copy <https://github.com/norment/moba_qc_imputation/blob/master/jobs/MERGE.job> to <ROOT> folder
2. Execute “sbatch MERGE.job” command.

The last step will convert from dosages (.bgen format) to hard calls (plink format), and merge INFO scores across imputation chunks.

## Merging hard calls across chromosomes

Submit the following script to merge the result across chromosomes:

<https://github.com/norment/moba_qc_imputation/blob/master/jobs/MERGE_PLINK_ALLCHR.job>

## Imputation report

At this point there are no step-by-step instructions to generate imputation report. It’s based on the following script:

<https://github.com/norment/moba_qc_imputation/blob/master/report/imputation_report_HCE.ipynb>, but it needs to be adjusted manually for each batch. A more recent version of these scripts is available here: <https://github.com/norment/moba_qc_imputation/blob/master/users/of/GSA_may2021/MoBa_pre_phasing_qc_and_imputation_reports.ipynb>