PRS Pipeline to Apply Weights from PGS Catalog v.0.9

M. Lin

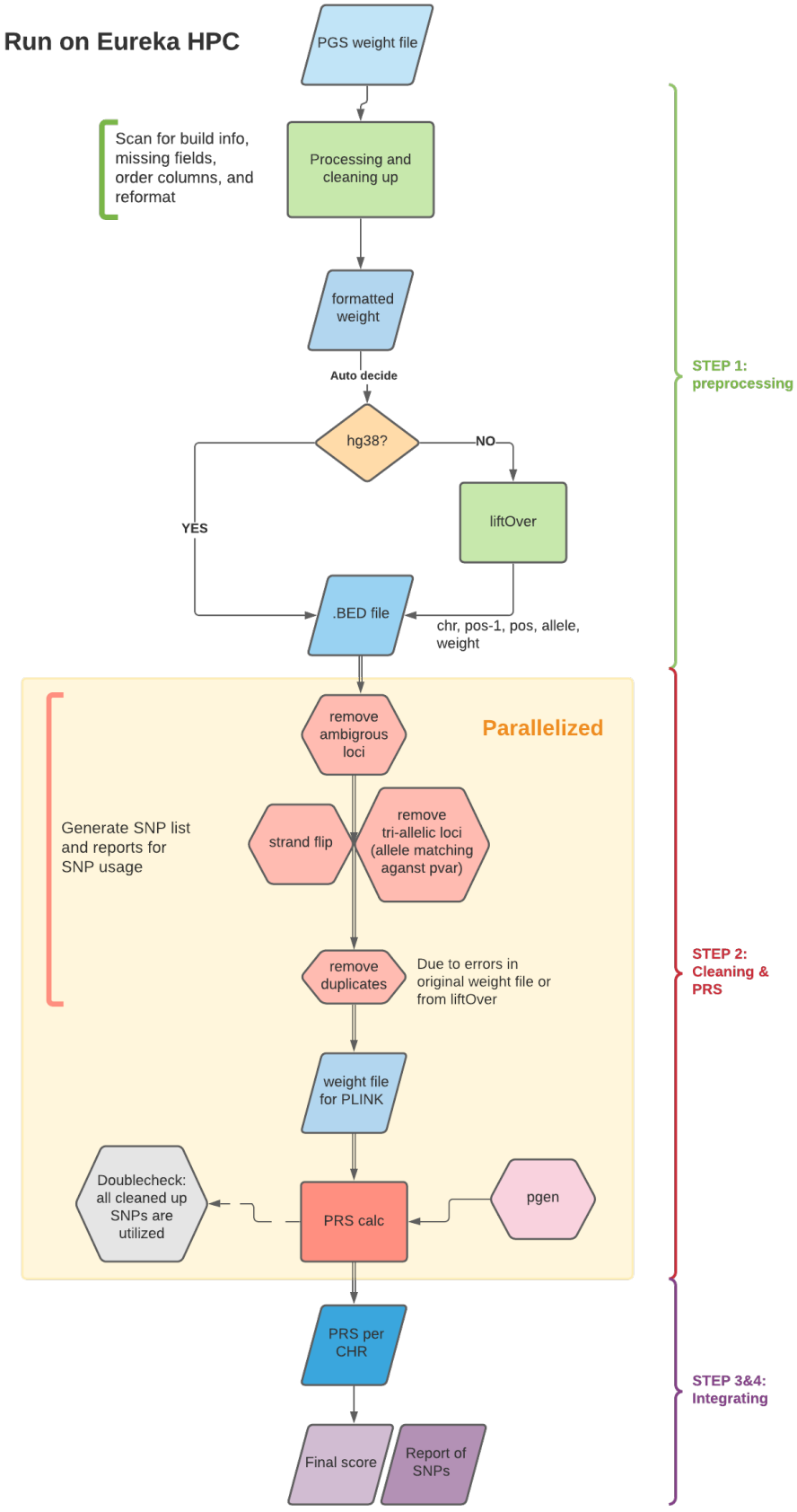
May 2022

General introduction of the pipeline

The PRS pipeline is stored on TIS google bucket. It takes in weight files in the format supported by PGS catalog, does proper QC against TOPMed imputed freeze 2 of CCPM biobank, computes PRS scores based on dosages of freeze 2, and generates short summary files to report the QC.

Relevant scripts are located in gs://hdchpcprodtis1-staging/mlin/scripts/pgs_pipeline/. When running the pipeline, the scripts also call external software such as PLINK and liftover, located in gs://hdchpcprodtis1-staging/mlin/bin/. There is **no** need to pre-copy any of the scripts or software into your space for the pipeline to work except for the master script, as the external files will be taken care of by the latter.

A general overview of the pipeline is visualized as below -


Run on Eureka HPC
PGS weight file
Scan for build info, missing fields, order columns, and reformat
Processing and cleaning up
formatted weight
Auto decide
hg38?
NO
YES
liftOver
.BED file
chr, pos-1, pos, allele, weight
STEP 1: preprocessing
Parallelized
remove ambigrous loci
Generate SNP list and reports for SNP usage
strand flip
remove tri-allelic loci (allele matching aganst pvar)
remove duplicates
Due to errors in original weight file or from liftOver
weight file for PLINK
Doublecheck: all cleaned up SNPs are utilized
PRS calc
pgen
STEP 2: Cleaning & PRS
PRS per CHR
Final score
Report of SNPs
STEP 3&4: Integrating

1. Explanations of the QC steps

The quality control process of the input file mostly aims to provide a list of variants and their weights that are concordant with CCPM freeze 2 data, which is imputed against TOPMed. The freeze 2 pgen file is stored at gs://hdchpcprodtis1-staging/mlin/freeze2_pgen/.

- Build liftover

Genome coordinates of chromosome and positions of variants in weights files will be lifted over to GRCh38/hg38, if not on this build already. Rather than matching the content of weight files against each build reference to detect the genome build of the input, the pipeline currently relies on an indication of build info in the header, such as # Original Genome Build or # genome_build (see the section of *Example of running the pipeline*)

The pipeline supports input files with builds on GRCH38/hg38, GRCh37/hg19, NCBI36/GRCh36/hg18, NCBI35/hg17.

- Converting rsID to genomic coordinates of hg38

If the input file has variant names in rsID, while no columns of chromosome or positions are provided, the pipeline will map to hg38 genomic coordinates to each variant based on dbSNP151 bed files.

- Removing ambiguous loci

Loci with risk & reference alleles that are A/T, C/G in the weight files will be removed.

- Removing duplicated loci that have the same risk allele, but each was assigned a different weight value

This has been a very rare observation, but it can happen from an error in the original weight files or liftOver process - duplicated loci with the same coordinates / names having the same allele codes, but the weight of the risk allele appears different.

- Strand matching with freeze 2

Strand flip will be performed on variants from the input file to match the same locus in freeze 2.

- Removing loci with mismatching allele code with freeze 2

If either strand flipping the input variant or not does not match with the allele code of the same locus in freeze 2 (e.g. a tri-allelic situation where input locus has allele A / G, freeze 2 has allele A / C), the variant will be removed. This is because all weight value of a risk allele is only relative to that of the correct reference allele, as estimated by initial association studies.

- Removing loci not found in freeze 2

2. Format of an input file

The input file is either a plain text file or gzipped text file (ending with .gz), providing weight and variant information in the format concordant with that in PGS catalog, as described here. The format supported by PGS catalog prior to early 2022 is slightly different, referred as **v1** here, compared to the latest format (**v2**). The pipeline accepts both, when clearly flagging version number.

In short, the pipeline requires basic fields of chromosome and bp positions and/or variant name in dbSNP rsID, effect allele, risk allele, weight of risk allele in beta (as opposed to OR). A header line with build info is also required. Irrelevant fields or headers will be ignored. Examples are provided as below (the order of columns does not matter):

- **Option 1 - provide rsID, risk and other alleles, and weight of risk allele**
  - **v1**

```
# Original Genome Build = GRCh37
rsID effect_allele reference_allele effect_weight
rs78540526 T C 0.1622
...
```

  - **v2**

```
# genome_build = hg37
rsID effect_allele other_allele effect_weight
rs7412746 C T -0.116
...
```

- **Option 2 - provide chromosome, bp, risk and other alleles, and weight of risk alleles**
  - **v1**

```
# Original Genome Build = hg38
chr_name chr_position effect_allele reference_allele effect_weight
11 69516650 T C 0.1622
...
```

  - **v2**

```
# genome_build = GRCh38
chr_name chr_position effect_allele other_allele effect_weight
11 69516650 T C 0.1622
...
```

- **Option 3 - provide rsID, chromosome, bp, risk and other alleles, and weight of risk alleles**
  - **v1**

```
# Original Genome Build = hg38
rsID chr_name chr_position effect_allele reference_allele effect_weight
rs78540526 11 69516650 T C 0.1622
...
```

– **v2**

```
# genome_build = GRCh38
rsID chr_name chr_position effect_allele other_allele effect_weight
rs78540526 11 69516650 T C 0.1622
...
```

3. Output files

3.1 PRS score of freeze 2 individuals

[prefix]_prs.sscore consists of two columns, with no headers. The first is IID of freeze 2. The second is raw PRS score.

3.2 Record of variants being used in PRS calculation for freeze 2

[prefix]_hg38_noAtCg_cleaned_forRecord.list contains a complete record of final variants being used for PRS calculation, after QC. Columns include:

CHR(hg38): Chromosome position of the variant, based on hg38

BP(hg38): Base pair position of the variant, based on hg38

OriginalSNPID: SNP ID from the input file, if not present, a chr:bp:a1:a2 matching freeze 2 will be assigned.

UpdatedSNPID: SNP ID with chr:bp:a1:a2 matching freeze 2

UpdatedRiskAllele: Risk allele code with strand matching freeze 2

UpdatedRefAllele: Reference/Other allele code with strand matching freeze 2

Weight: Weight value of the updated risk allele

3.3 Variants discarded / changed during QC

These files will be stored in a sub-directory recordfiles/ nested under the user-specified bucket directory.

[prefix]_hg38_noAtCg_flipped.list records variants that are used but with strand flipped to match freeze 2.

[prefix]_hg38_noAtCg_mismatch.list records variants with allele codes not matching freeze 2, and thus are not included in final PRS calculation.

[prefix]_hg38_noAtCg_missing_in_pvar.list records variants that are not present in freeze 2, and thus are not included in final PRS calculation.

3.4 Log file

[prefix]_prs.log contains a summary of counts of variants during QC, and change of build (i.e. liftover). An example of log file:

```
Total number of input weight file: 52.
Lifting over from hg19 to hg38.
Discarded unmatched variants from liftOver: 0
```

Number of non-autosomal variants being discarded: 0

Number of ambigous A/T, C/G loci that are removed: 6

Number of weight SNPs that are flipped to the other strand: 2

Number of SNPs with mismatched allele codes against pgen that are removed: 0

Number of SNPs in weight file not found in pgen that are removed: 1

Number of duplicated entries with same position and alleles (and are removed): 0

Number of final variants used for PRS: 45

4. Example of running the pipeline

The master script **masterPRS_format_v2.sh** is the only script that needs to be copied to a working directory prior to running to pipeline. It takes in five parameters -

bash masterPRS_format_v2.sh [input version#] [bucket directory for the input file] [input file name] [bucket directory to store the outputs] [output prefix]

[input version#] refers to the different version of **v1** or **v2** format from PGS catalog as described above, only "1" or "2" is needed as input here.

If the last parameter,*[output prefix]*, is put as "unknown", then the prefix of the output files will be auto-determined by scanning for header lines of # PGS ID =, or # pgs_id = (only when # format_version=2.0 is also present). Otherwise, it will be the prefix name provided by user.

An example of using slurm to run the master script on eureka HPC -

```
#!/bin/sh
#SBATCH -p c2s8
#SBATCH --ntasks=1
#SBATCH --job-name=PRS
#SBATCH --error=PRS.err
#SBATCH --out=PRS.out

# staging in data and container
set -e # fail script on any error
starttime=`date +%s`
DIR=$(mktemp -d)
cd $DIR

# cp the script
gsutil -q cp gs://hdchpcprodtis1-staging/mlin/scripts/pgs_pipeline/masterPRS_format_v2.sh .
chmod 700 ./*

# run
bash masterPRS_format_v2.sh 2 \
    gs://hdchpcprodtis1-staging/mlin/pgs_scores/pigmentation \
    PGS002155.txt.gz \
    gs://hdchpcprodtis1-staging/dump \
    pigmentation


endtime=`date +%s`
```

```
runtime=$((endtime - starttime))
echo "Total run time of this pipeline is ${runtime}s"
```

For other questions, please contact meng.lin@cuanschutz.edu