PRS harmonization and calculation pipeline to apply weights to genetic data

ESCALATOR container v1.00

Meng Lin and Matthew Fisher

Mar 2024

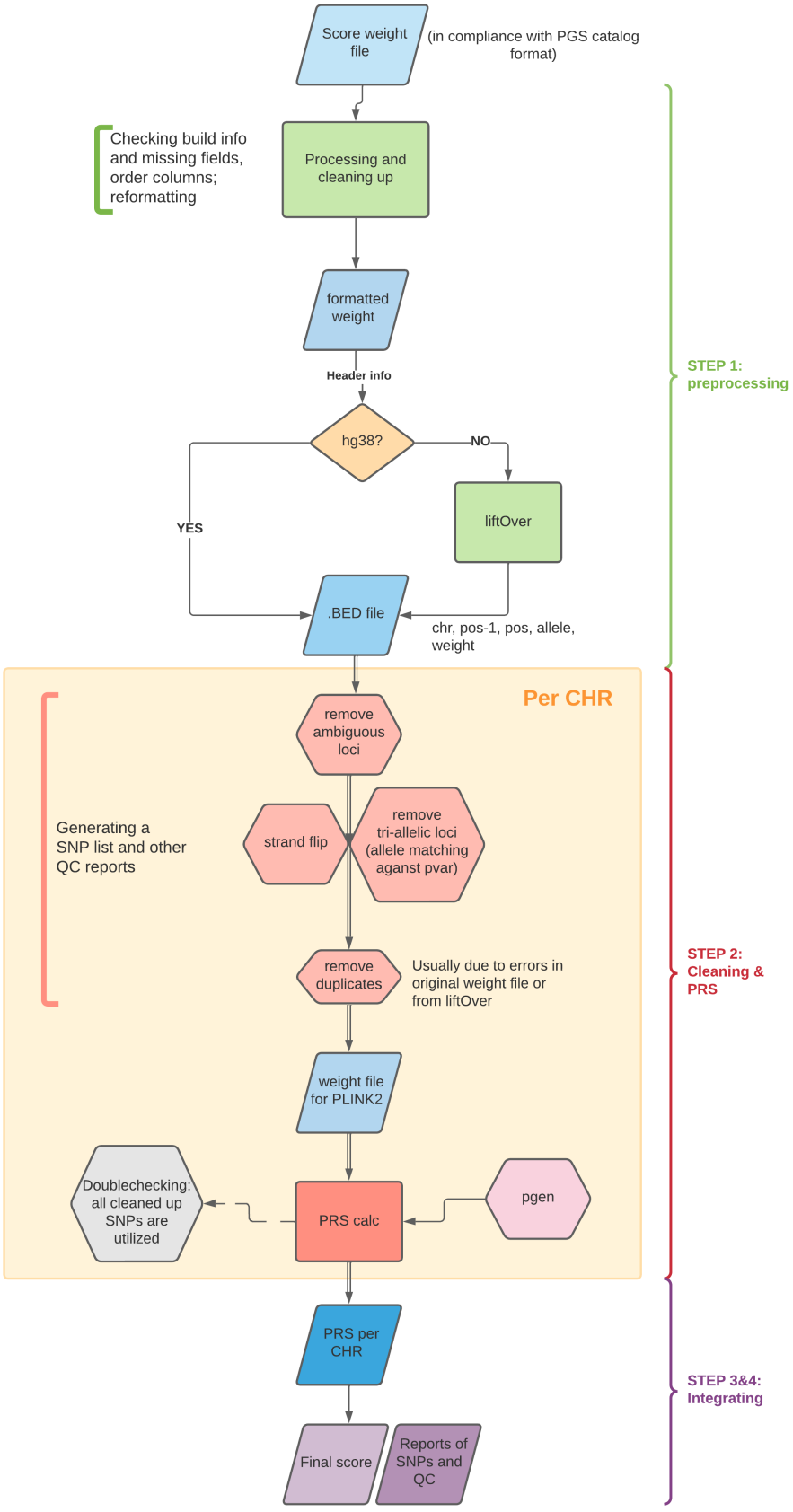
General introduction of the pipeline

This is a containerized PRS harmonization pipeline taken from the original pipeline scripts used on Eureka cloud platform at CU Anschutz (https://github.com/menglin44/escalator). Generalized scripts adapting non-Eureka environment have been under development and test, for those scripts and if you would like to build the ESCALATOR container image on your own, please visit https://github.com/MatthewFisher126/ESCALATOR.

The pipeline takes in weight files in the format supported by PGS catalog, performs harmonization/QC against a target genetic dataset provided by users (in PLINK v2. binary pfile format), computes PRS for each sample in the dataset, and generates short summary files to report the QC process. See the following sections for accepted formats and examples of how to run the container.

- Some important general pre-assumptions of the current version of the pipeline include -
  - Singularity is already installed in your computational environment.
  - Your test dataset (i.e. genetic data of the samples you wish scores to be calculated on) is on build GRCh38. Weight files can be on any other build, then the weights not on GRCh38 will be lifted by ESCALATOR to this build before being harmonized against the test dataset loci.
  - Your test dataset is parsed by chromosomes, from 1-22 (i.e. autosomes). The parsed files are named with "chr" followed by chromosome number and then the general infix of the file, e.g. chr1_myfile.pvar, chr1_myfile.psam, chr1_myfile.pgen.
  - The containerized pipeline does not include the original function of fetching chromosome and bp position info from dbSNP variant names. Thus the weight file will be expected to contain correct chromosomal and position info itself.
- Please note that **the output score values generated by PLINK2 score function used in ESCALATOR is halved (see here)**, as PLINK2 divides scores $a^T G$ by the number of allele observations. In the majority of PRS-related analyses, this does not impact the final results, as scores need to be standardized or scaled before use. Yet if other methods of using scores is involved, such as summing two scores that require them to be on the same scale to reflect the raw weight per variant, users should be aware of how PLINK handles the averaged score and consider multiplying the score values by two.

A general overview of the pipeline is visualized as below -


Score weight file
(in compliance with PGS catalog format)
Checking build info and missing fields, order columns; reformatting
Processing and cleaning up
formatted weight
Header info
hg38?
NO
YES
liftOver
.BED file
chr, pos-1, pos, allele, weight
STEP 1: preprocessing
Per CHR
remove ambiguous loci
Generating a SNP list and other QC reports
strand flip
remove tri-allelic loci (allele matching aganst pvar)
remove duplicates
Usually due to errors in original weight file or from liftOver
weight file for PLINK2
Doublechecking: all cleaned up SNPs are utilized
PRS calc
pgen
STEP 2: Cleaning & PRS
PRS per CHR
Final score
Reports of SNPs and QC
STEP 3&4: Integrating

1. Explanations of the QC steps

- 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.

- 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 the test dataset

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

- Removing loci with mismatching allele code with the test dataset

If either strand flipping the input variant or not does not match with the allele code of the same locus in the test dataset (e.g. a tri-allelic situation where input locus has allele A / G, the test dataset 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 the loci not found in the test dataset

2. Score calculation after harmonization

ESCALATOR leverages external software of PLINK2, which is included in the container.

3. 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, 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
...
```

4. Output files

4.1 A list of PRS score of all samples in the test dataset

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

4.2 A record of variants being used in PRS calculation

[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, or variant name, from the input file, if not present, a chr:bp:ref:alt matching the test dataset will be assigned.

UpdatedSNPID: SNP ID, or variant name, used in the test dataset

UpdatedRiskAllele: Risk allele code with strand matching the test dataset

UpdatedRefAllele: Reference/Other allele code with strand matching the test dataset

Weight: Weight value of the updated risk allele

4.3 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 the test dataset.

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

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

4.4 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

4.5 In situations of unsuccessful runs

A PRS might not be calculated successfully in the following situations, and would have a message generated in the last line of the corresponding log file.

- Genome build is not listed / unclear

```
Didn\'t detect the genome build of the input file
```

- Some of the essential column information is missing in the original weight file (see section 2)

```
Reformatting of the input file failed
```

- No variants are left in the weight file after QC / none of the variants are found the in target pfile

```
No variants are left for PRS score
```

5. Example of running the pipeline

```
singularity exec escalator-v1.sif masterPRS_format_v2_freeze3.sh 2 \
    /home/user/score_weights \
    PGS002155.txt.gz \
    /home/user/score_output \
    pigmentation \
    /home/user/genetic_data_plink \
    freeze3_imputed_dosages
```

The general command with required arguments used above is:

singularity exec escalator-v1.sif masterPRS_format_v2_freeze3.sh [format version of weight] \

[weight file directory] \

[weight file] \

[output directory] \

[output prefix] \

[test dataset directory] \

[test dataset infix]

- Explanations of each argument:

  1. *[format version of weight file]* refers to the different version of **v1** or **v2** format from PGS catalog as Section 3.
     - Only "1" or "2" is needed as input here. (If other characters other than 1 or 2 are used, the script will search for a relevant header 'format_version=2.0' to assume v2, or otherwise v1.
  2. *[weight file directory]* and *[weight file]*: The directory and file name of the score weight file.
  3. *[output directory]*: The directory under which the calculated score, and the other records files will be written.
  4. *[output prefix]*: Prefix of the output file names.
     - For example, it often can be the name of the phenotype ("pigmentation") or score number from the PGS catalog (e.g. "PGS00001").
     - In case of this argument being specified as "unknown", the prefix of the output files will be determined by ESCALATOR scanning for header lines of # PGS ID =, or # pgs_id = (only when # format_version=2.0 is also present).
  5. *[test dataset directory]*: The directory where the genetic dataset of samples of interest rests.
  6. *[test dataset infix]*: The infix of the genetic dataset, parsed by each chromosome from 1-22.
     - E.g. An infix of "myfile" will let the pipeline automatically search for *chr1_myfile.psam, chr1_myfile.pvar, chr1_myfile.pgen* . . . all the way to *chr22_myfile.psam, chr22_myfile.pvar, chr22_myfile.pgen* in the specified test dataset directory.

Alternatively, to see general help information from the container, type -

```
singularity run-help escalator-v1.sif
```

6. Miscellaneous

If your original files are in VCF format and do not come with variant names to be converted to PLINK2 file (for example, all variant names are "." in 1000 Genomes low coverage as below),

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  HG00096
22      10516173        .       A       G       .       PASS    AC=121;AN=5096;DP=8203;AF=0.02;EAS_AF=0;EUR_AF=0.02;AFR_AF=0.06;AMR_AF=0.02;SAS_AF=0;VT=SNP;NS=2548     GT      0|0
22      10522217        .       G       A       .       PASS    AC=89;AN=5096;DP=9085;AF=0.02;EAS_AF=0;EUR_AF=0;AFR_AF=0.07;AMR_AF=0;SAS_AF=0;VT=SNP;NS=2548    GT      0|0
22      10526445        .       A       G       .       PASS    AC=4948;AN=5096;DP=7978;AF=0.97;EAS_AF=0.98;EUR_AF=1;AFR_AF=0.93;AMR_AF=0.97;SAS_AF=1;VT=SNP;NS=2548    GT      1|1
22      10527034        .       G       T       .       PASS    AC=271;AN=5096;DP=5974;AF=0.05;EAS_AF=0.01;EUR_AF=0.03;AFR_AF=0.11;AMR_AF=0.03;SAS_AF=0.07;VT=SNP;NS=2548       GT      0|0
22      10527038        .       C       G       .       PASS    AC=267;AN=5096;DP=5947;AF=0.05;EAS_AF=0.08;EUR_AF=0.03;AFR_AF=0.01;AMR_AF=0.07;SAS_AF=0.08;VT=SNP;NS=2548       GT      0|0
```

you can use the extra flags `--set-all-var-ids chr@:#:\$r:\$a --new-id-max-allele-len 150`

such in the example of

```
for i in {1..22}
do
./plink2 --vcf ALL.chr${i}.shapeit2_integrated_snvindels_v2a_27022019.GRCh38.phased.vcf.gz \
--set-all-var-ids chr@:#:\$r:\$a --new-id-max-allele-len 150 \
--make-pgen --out chr${i}_tgp_snvindels
done
```

when converting VCF to PLINK2 file.

For other questions, please contact *meng.lin@cuanschutz.edu* and *matthew.j.fisher@cuanschutz.edu*