Guide to the process of using user source files

Original data file

This *Precision Medicine Database* is a pharmacogenomics knowledge resource that encompasses clinical information including clinical guidelines and drug labels, potentially clinically actionable gene-drug associations and genotype-phenotype relationships. And our database collects, curates and disseminates knowledge about the impact of human genetic variation on drug responses through the *PharmGKB*.

The actual information that is retrieved or queried in our database, such as Variant ID, is obtained by extracting the relevant information from the processed data. An evaluation of existing user data formats revealed that the majority of users use the FASTQ format, a text format that stores biological sequences and corresponding quality evaluations, which is almost the standard format for high-throughput sequencing, and which determines its prevalence of use. Since our database cannot support online FASTQ data format to VCF data format processing, we present this data processing flow guide for users' reference.

Data format conversion: from FASTQ to VCF

This guide uses a single sample of PE100 as an example, SE and multiple sample data (sequencing multiple samples from the same batch, generating multiple FASTQ files for common analysis) will be described separately in the guide.

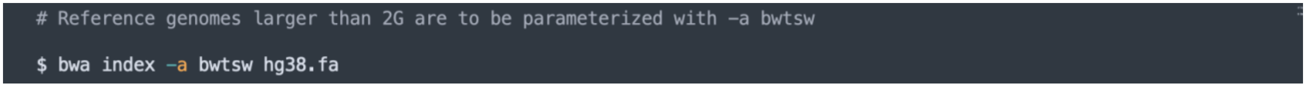
PS: The command line and process in this guide are for reference only, please deal with the specific situation according to your own data strain. Each software version may have slightly different names of parameters, if there is a misunderstanding of parameters, please check the parameter directory of the current version with -h and find the corresponding parameters to set.

All required software can be installed through the conda environment:
https://anaconda.org/bioconda

1. Data Preparation

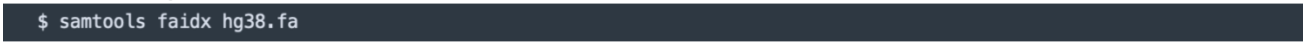
The reference genome was selected as hg38, and the reference genome is available for download at: https://hgdownload.cse.ucsc.edu/goldenPath/hg38/bigZips/

Reference file indexing: Do not use nohup for this step, because the output data will be wrong. Here, take hg38 as an example, just index it once.



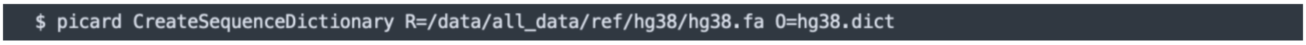
```
# Reference genomes larger than 2G are to be parameterized with -a bwtsw

$ bwa index -a bwtsw hg38.fa
```

Generate hg38.fasta.fai files:



```
$ samtools faidx hg38.fa
```

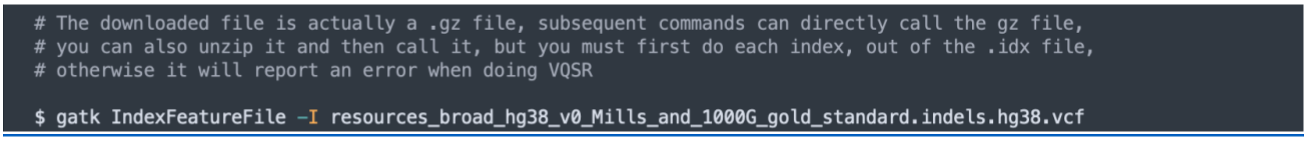
Generate dict file of reference genome:



```
$ picard CreateSequenceDictionary R=/data/all_data/ref/hg38/hg38.fa O=hg38.dict
```

2. gatk4 reference file preparation

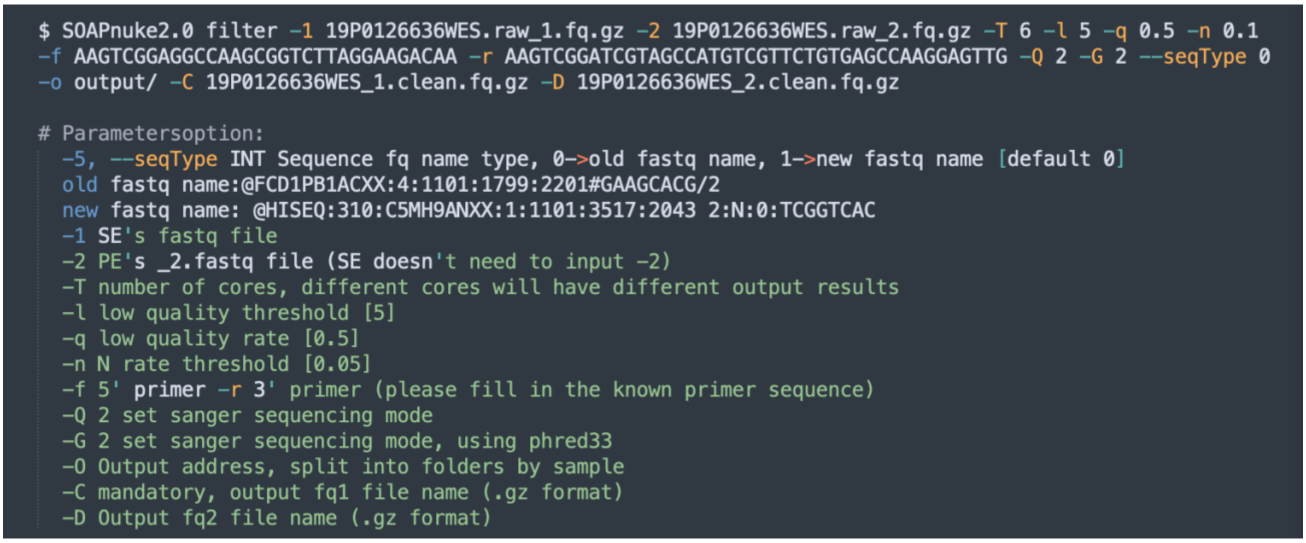
gatk4 reference file downloaded:
https://console.cloud.google.com/storage/browser/genomics-public-data/resources/broad/hg38/v0/



```
# The downloaded file is actually a .gz file, subsequent commands can directly call the gz file,
# you can also unzip it and then call it, but you must first do each index, out of the .idx file,
# otherwise it will report an error when doing VQSR

$ gatk IndexFeatureFile -I resources_broad_hg38_v0_Mills_and_1000G_gold_standard.indels.hg38.vcf
```

3. Raw data quality control

Softwires needed: SOAPnuke, Trimmomatic, fastqc
Filtering operation on reads with software: de-adaptor, index, low quality bases and reads removal, high percentage of N reads removal.



```
$ SOAPnuke2.0 filter -1 19P0126636WES.raw_1.fq.gz -2 19P0126636WES.raw_2.fq.gz -T 6 -l 5 -q 0.5 -n 0.1
-f AAGTCGGAGGCCAAGCGGTCTTAGGAAGACAA -r AAGTCGGATCGTAGCCATGTCGTTCTGTGAGCCAAGGAGTTG -Q 2 -G 2 --seqType 0
-o output/ -C 19P0126636WES_1.clean.fq.gz -D 19P0126636WES_2.clean.fq.gz

# Parametersoption:
  -5, --seqType INT Sequence fq name type, 0->old fastq name, 1->new fastq name [default 0]
  old fastq name:@FCD1PB1ACXX:4:1101:1799:2201#GAAGCACG/2
  new fastq name: @HISEQ:310:C5MH9ANXX:1:1101:3517:2043 2:N:0:TCGGTCAC
  -1 SE's fastq file
  -2 PE's _2.fastq file (SE doesn't need to input -2)
  -T number of cores, different cores will have different output results
  -l low quality threshold [5]
  -q low quality rate [0.5]
  -n N rate threshold [0.05]
  -f 5' primer -r 3' primer (please fill in the known primer sequence)
  -Q 2 set sanger sequencing mode
  -G 2 set sanger sequencing mode, using phred33
  -O Output address, split into folders by sample
  -C mandatory, output fq1 file name (.gz format)
  -D Output fq2 file name (.gz format)
```

Check fastq quality with fastqc, continue processing if you fail, and go directly to the next step if you PASS.

According to fastqc results, use Trimmomatic to remove XX bp (this step on-demand, this software has many other parameters, can directly replace SOAPnuke for filtering operations, better to make also more time-consuming)

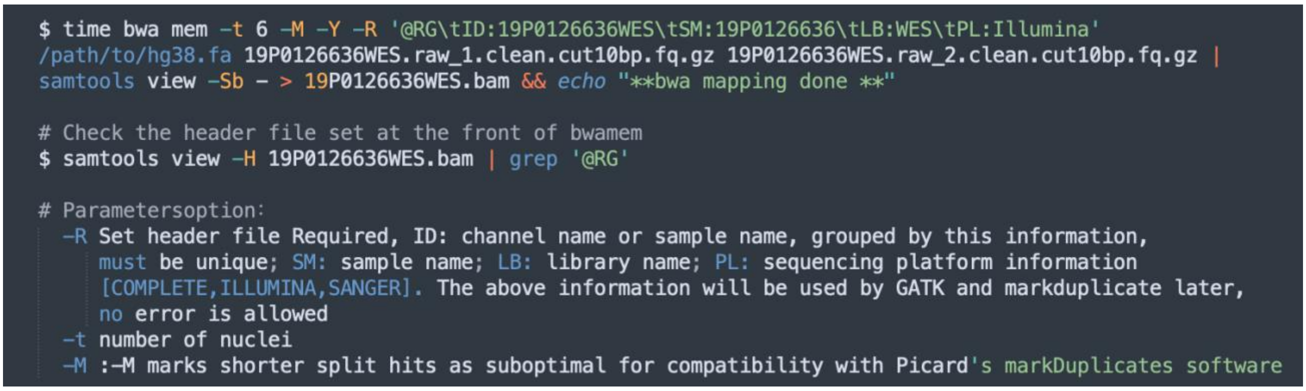
4. Sequence Matching

Software needed: bwa mem

If there are 4 fastqs in a sample, the FASTQ files are processed separately here, and the two fq files of the same channel are processed together to generate a sam/bam file, so that a total of 2 sam/bam files are generated for a sample, and the merge step is performed later.

If there are only 2 fastq files for a sample, one bam file is generated for each sample.

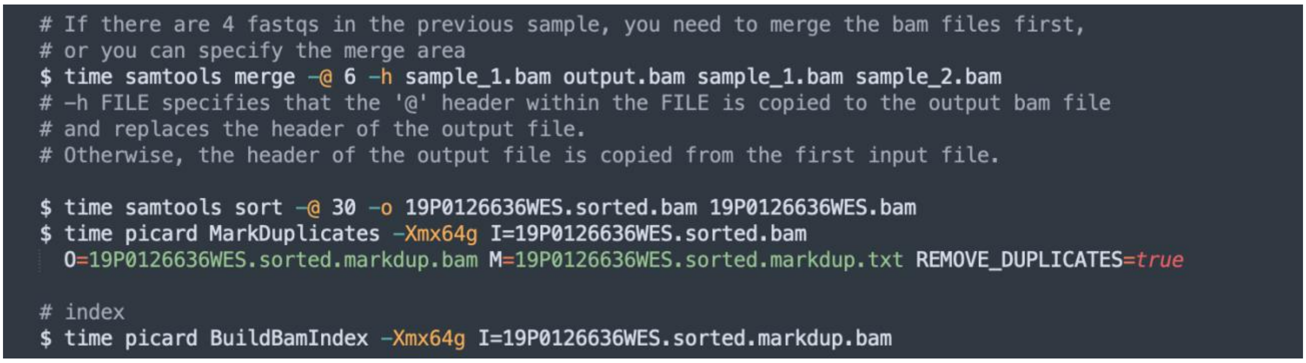
The -R parameter is very important for the subsequent steps, if you set it wrong, you need to run this step again.



```
$ time bwa mem -t 6 -M -Y -R '@RG\tID:19P0126636WES\tSM:19P0126636\tLB:WES\tPL:Illumina'
/path/to/hg38.fa 19P0126636WES.raw_1.clean.cut10bp.fq.gz 19P0126636WES.raw_2.clean.cut10bp.fq.gz |
samtools view -Sb - > 19P0126636WES.bam && echo "**bwa mapping done **"

# Check the header file set at the front of bwamem
$ samtools view -H 19P0126636WES.bam | grep '@RG'

# Parametersoption:
  -R Set header file Required, ID: channel name or sample name, grouped by this information,
     must be unique; SM: sample name; LB: library name; PL: sequencing platform information
     [COMPLETE,ILLUMINA,SANGER]. The above information will be used by GATK and markduplicate later,
     no error is allowed
  -t number of nuclei
  -M :-M marks shorter split hits as suboptimal for compatibility with Picard's markDuplicates software
```

5. sam/bam file pre-processing

Softwires needed: samtools, picard (based on java)

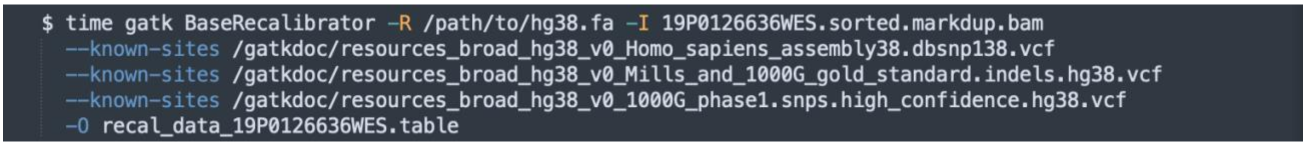


```
# If there are 4 fastqs in the previous sample, you need to merge the bam files first,
# or you can specify the merge area
$ time samtools merge -@ 6 -h sample_1.bam output.bam sample_1.bam sample_2.bam
# -h FILE specifies that the '@' header within the FILE is copied to the output bam file
# and replaces the header of the output file.
# Otherwise, the header of the output file is copied from the first input file.

$ time samtools sort -@ 30 -o 19P0126636WES.sorted.bam 19P0126636WES.bam
$ time picard MarkDuplicates -Xmx64g I=19P0126636WES.sorted.bam
  O=19P0126636WES.sorted.markdup.bam M=19P0126636WES.sorted.markdup.txt REMOVE_DUPLICATES=true

# index
$ time picard BuildBamIndex -Xmx64g I=19P0126636WES.sorted.markdup.bam
```

6. BQSR (Recalibration Base Quality Score)

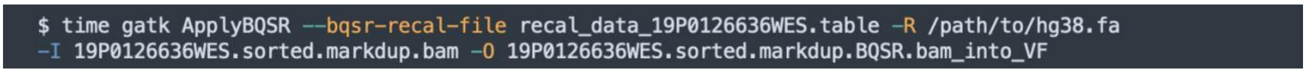
The so-called variant loci are the parts of the genome that are different from the reference genome. Assuming that there are some systematic errors due to the sequencing instrumentation in the original data, the variant found during the variant identification process will have many false positives.
The main purpose of this step is to adjust the mass fraction of the original bases.

Using the existing snp database, a correlation model was developed to generate a recalibration table, which was entered into the database of known polymorphic loci and used to mask those parts that did not require recalibration.



```
$ time gatk BaseRecalibrator -R /path/to/hg38.fa -I 19P0126636WES.sorted.markdup.bam
  --known-sites /gatkdoc/resources_broad_hg38_v0_Homo_sapiens_assembly38.dbsnp138.vcf
  --known-sites /gatkdoc/resources_broad_hg38_v0_Mills_and_1000G_gold_standard.indels.hg38.vcf
  --known-sites /gatkdoc/resources_broad_hg38_v0_1000G_phase1.snps.high_confidence.hg38.vcf
  -O recal_data_19P0126636WES.table
```

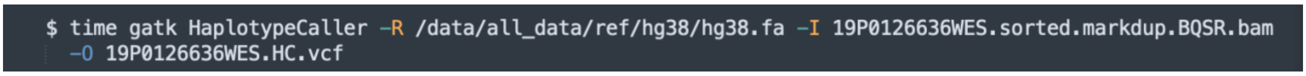
Adjustment of the original bases according to this model will only adjust non-known SNP regions.



```
$ time gatk ApplyBQSR --bqsr-recal-file recal_data_19P0126636WES.table -R /path/to/hg38.fa
-I 19P0126636WES.sorted.markdup.bam -O 19P0126636WES.sorted.markdup.BQSR.bam_into_VF
```

7. Generate VCF files

Use HaplotypeCaller to call variant on the above bam file, which is the process of finding variant and generating a VCF file for subsequent loci quality control and annotation, from a bam to a VCF file.

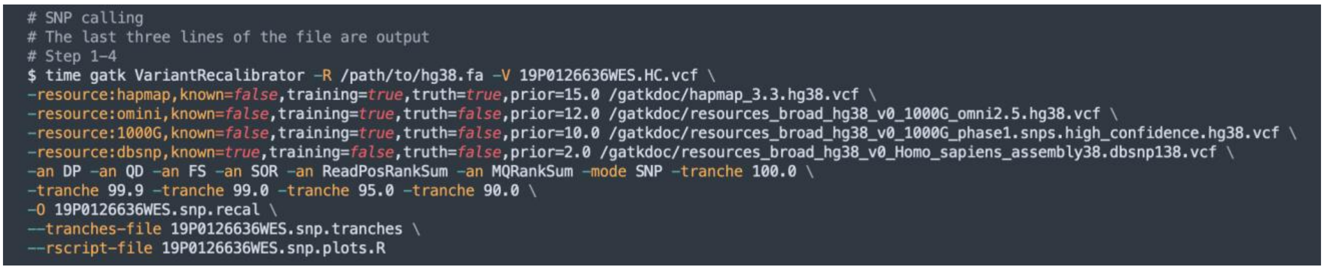


```
$ time gatk HaplotypeCaller -R /data/all_data/ref/hg38/hg38.fa -I 19P0126636WES.sorted.markdup.BQSR.bam
  -O 19P0126636WES.HC.vcf
```

8. Variation quality control

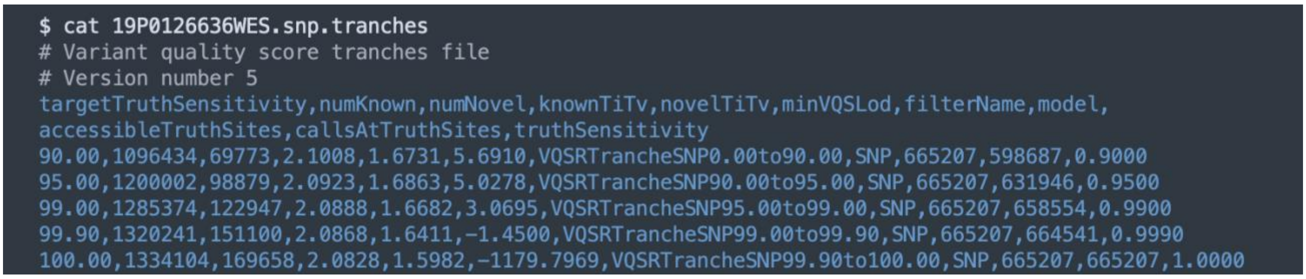
GATK VQSR was used for quality control of variation data to eliminate false positive results as much as possible.
The VQSR refers to six indicator thresholds, which are:

[1] QualByDepth (QD)
[2] FisherStrand (FS)
[3] StrandOddsRatio (SOR)
[4] RMSMappingQuality (MQ)
[5] MappingQualityRankSumTest (MQRankSum)
[6] ReadPosRankSumTest (ReadPosRankSum)



```
# SNP calling
# The last three lines of the file are output
# Step 1-4
$ time gatk VariantRecalibrator -R /path/to/hg38.fa -V 19P0126636WES.HC.vcf \
-resource:hapmap,known=false,training=true,truth=true,prior=15.0 /gatkdoc/hapmap_3.3.hg38.vcf \
-resource:omini,known=false,training=true,truth=false,prior=12.0 /gatkdoc/resources_broad_hg38_v0_1000G_omni2.5.hg38.vcf \
-resource:1000G,known=false,training=true,truth=false,prior=10.0 /gatkdoc/resources_broad_hg38_v0_1000G_phase1.snps.high_confidence.hg38.vcf \
-resource:dbsnp,known=true,training=false,truth=false,prior=2.0 /gatkdoc/resources_broad_hg38_v0_Homo_sapiens_assembly38.dbsnp138.vcf \
-an DP -an QD -an FS -an SOR -an ReadPosRankSum -an MQRankSum -mode SNP -tranche 100.0 \
-tranche 99.9 -tranche 99.0 -tranche 95.0 -tranche 90.0 \
-O 19P0126636WES.snp.recal \
--tranches-file 19P0126636WES.snp.tranches \
--rscript-file 19P0126636WES.snp.plots.R
```

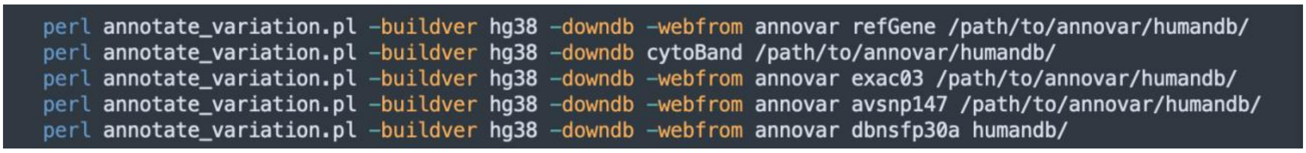
View data files



```
$ cat 19P0126636WES.snp.tranches
# Variant quality score tranches file
# Version number 5
targetTruthSensitivity,numKnown,numNovel,knownTiTv,novelTiTv,minVQSLod,filterName,model,
accessibleTruthSites,callsAtTruthSites,truthSensitivity
90.00,1096434,69773,2.1008,1.6731,5.6910,VQSRTrancheSNP0.00to90.00,SNP,665207,598687,0.9000
95.00,1200002,98879,2.0923,1.6863,5.0278,VQSRTrancheSNP90.00to95.00,SNP,665207,631946,0.9500
99.00,1285374,122947,2.0888,1.6682,3.0695,VQSRTrancheSNP95.00to99.00,SNP,665207,658554,0.9900
99.90,1320241,151100,2.0868,1.6411,-1.4500,VQSRTrancheSNP99.00to99.90,SNP,665207,664541,0.9990
100.00,1334104,169658,2.0828,1.5982,-1179.7969,VQSRTrancheSNP99.90to100.00,SNP,665207,665207,1.0000
```

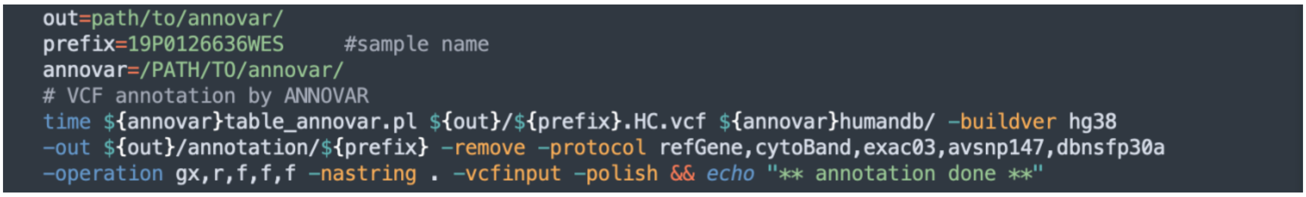
9. Site comments

We use ANNOVAR software for site annotation.
You need to download the database comment file in annovar/humandb/ (for example, HG38).



```
perl annotate_variation.pl -buildver hg38 -downdb -webfrom annovar refGene /path/to/annovar/humandb/
perl annotate_variation.pl -buildver hg38 -downdb cytoBand /path/to/annovar/humandb/
perl annotate_variation.pl -buildver hg38 -downdb -webfrom annovar exac03 /path/to/annovar/humandb/
perl annotate_variation.pl -buildver hg38 -downdb -webfrom annovar avsnp147 /path/to/annovar/humandb/
perl annotate_variation.pl -buildver hg38 -downdb -webfrom annovar dbnsfp30a humandb/
```

The following command goes back to our working directory (where we've been doing Variant Calling), so I added ${annovar} to the pl file just to call this program. Run the comment command.



```
out=path/to/annovar/
prefix=19P0126636WES     #sample name
annovar=/PATH/TO/annovar/
# VCF annotation by ANNOVAR
time ${annovar}table_annovar.pl ${out}/${prefix}.HC.vcf ${annovar}humandb/ -buildver hg38
-out ${out}/annotation/${prefix} -remove -protocol refGene,cytoBand,exac03,avsnp147,dbnsfp30a
-operation gx,r,f,f,f -nastring . -vcfinput -polish && echo "** annotation done **"
```

Finally, we have annotated VCF files.