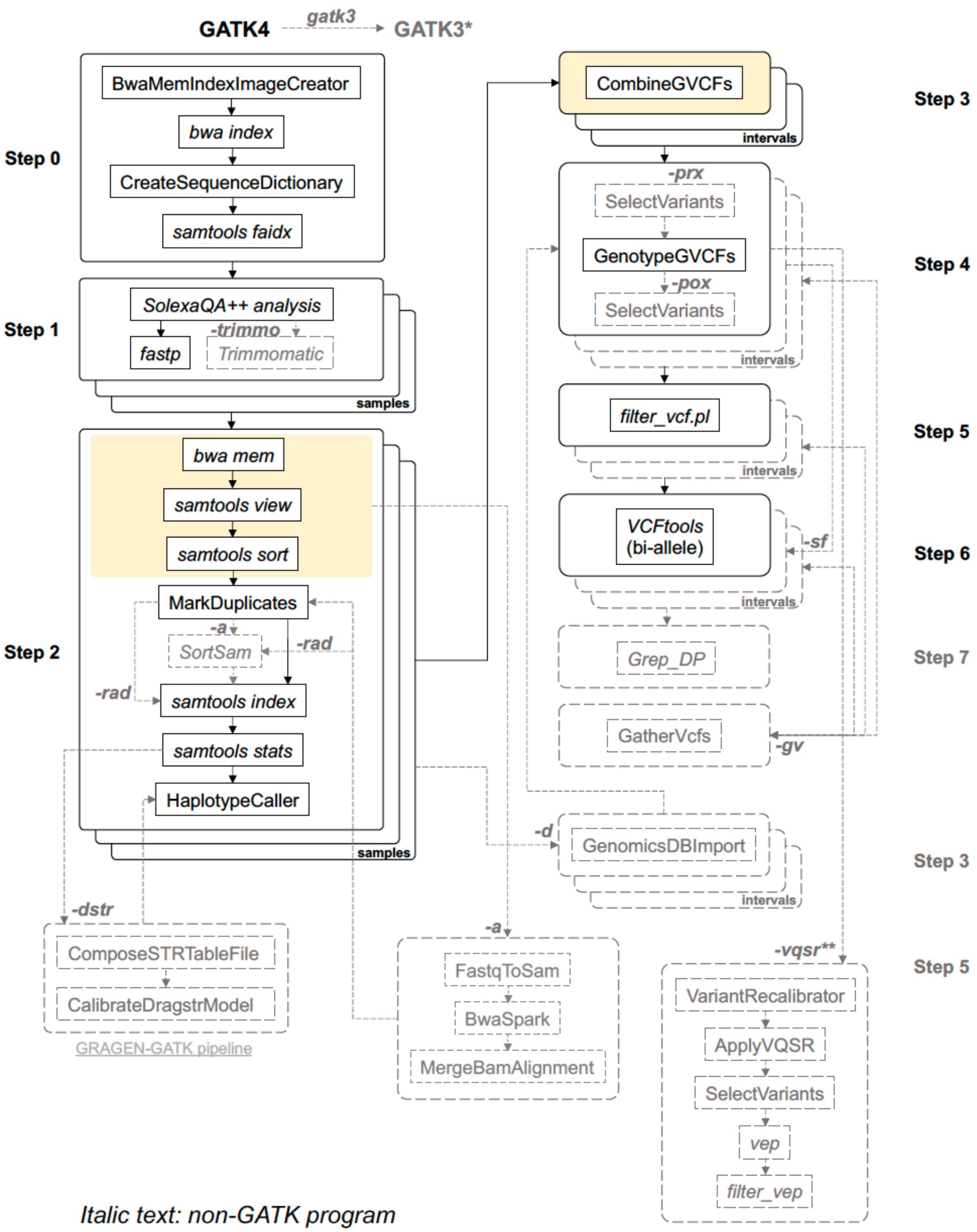
Manual of GATK_pipeline.pl v.3.4

Please make sure **qsub_subroutine.pl** is in $HOME/softwares. If not, please put the file in.
Please check if all the environmental settings are set. (Page 11)

Pipeline overview



Italic text: non-GATK program

*GATK3 pipeline is only recommanded for gvcf entry files created by GATK3.
**VQSR pipeline: you need to download databases, set R environment, and VEP environment

Basic usage:

For use of GATK4 pipeline:

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -g GROUP_NAME –exc [optional arguments]
```

For use of GATK3 pipeline (the same as old pipeline):

```
perl GATK_pipeline.pl gatk3 -p PATH -r REFERENCE_FILE -g GROUP_NAME -exc [optional arguments]
```

Advanced usage:

Single step running (eg. Step 3):

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -sp 3s –exc [optional arguments]
```

Multiple steps running (eg. From step 1 to step 4):

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -g GROUP_NAME -sp 1p -esp 4 –exc [optional arguments]
```

Pseudo-running (only generate command lines but does not send jobs):

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -g GROUP_NAME [optional arguments]
```

Using RAD (DArT) data:

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -g GROUP_NAME -rad –exc [optional arguments]
```

Overwrite step 3 files:

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -sp 3s -ow –exc [optional arguments]
```

Using user defined input sample list in step 3:

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -sp 3s --list LIST_FILE_PATH –exc [optional arguments]
```

Generate vcf(s) of all non-variant sites:

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -sp 4s -as –exc [optional arguments]
```

Pre-select vcf before GenotypeGVCFs:

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -sp 4s -prx SAMPLE_LIST.arg –exc [optional arguments]
```

Using pre-selected vcfs for GenotypeGVCFs:

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -sp 4s -ps SELECTED_VCF_FOLDER_PATH –exc [optional arguments]
```

Using the same sample run results as previous runs (eg. SN: ABCD):

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -sn ABCD –exc [optional arguments]
```

Gathering chromosomal vcfs into one vcf (step 4 ,5 ,6):
(run step 4, gathering at step 4)

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -sp 4s -gv 4 –exc [optional arguments]
```

(start from step 3, gathering at step 5)

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -g GROUP_NAME -sp 3p -gv 5 –exc [optional arguments]
```

Start from step 6 with raw vcf:

```
Perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE –sp 6s -sf -exc
```

NOTE: -sf should always be used when using raw vcfs as inputs.

Run the pipeline locally (highly NOT recommended):

```
perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -g GROUP_NAME -lc
```

IMPORTANTFor background running (Recommended in ***P*** mode):

```
nohup perl GATK_pipeline.pl -p PATH -r REFERENCE_FILE -g GROUP_NAME –exc [optional arguments] > Log.txt 2>&1 &
```

#This command line will run perl script in background, and put STDOUT and STDERR into Log.txt file.
#The pipeline will NOT be terminated when you turn off the terminal.
#You can use "tail -f Log.txt" to look at the running process.
#The qsub jobs are detected automatically by the script. However, if you want to check it manually, just use the qstat command.

This script can check necessary files in every step, and continuously run from the available files. Thus, **if you accidentally stop the pipeline from running, you can use the same command line to run again (with -*sn* argument).
However, if some existing files are incomplete, the pipeline may stop. You can simply run the "**Check_log.pl**" script and delete the incomplete file and run again. (Please check the log file of each job, the pipeline will remind you if there are any error records in the log file after running.)

IMPORTANT**filter_vcf_2.4.pl** file is required for filtering step, don't forget to put the file in the same root with GATK_pipeline.pl

IMPORTANTGATK4 pipeline is recommended. There are many bugs in GenotypeGVCFs in GATK3, especially when you have many large gvcfs (~1Gb*150 samples) to do. However, don't use gvcf generated from GATK3 to run GATK4 pipeline, it causes problems. You can start from GATK3 generated bam files.

IMPORTANTIf you run ***p*** mode through step 2, ***-g*** is required.

IMPORTANTIf your reference file has many contigs (chromosomes/intervals), please use -ns argument. This will NOT separate contigs into different files.

IMPORTANTIf VQSR filtering is used, you need to make sure that the R environment and a R package, ggplot2, are installed. The VEP software should be installed as well. Installation of VEP by Conda environment is highly recommended. You can setup the R and VEP environment by edition of the line 80 and 81 in the GATK_pipeline_v3.4.pl. VEP:$vep_env, R:$r_env.

If you will run pipeline through step 7, and the prefix of the contig you want to grep is not "chr", please indicate the correct prefix with *-pfdp***.

By default, CombineGVCFs will be used in step 3. You can switch to GenomicsDBImport by *-d*** argument. If you have sample number larger than 1500, ***-d*** is recommended.

**If you want to kill all the running qsub jobs in the specific SN, use following command: qstat -u *USERNAME* | grep "*SN*" | cut -d"." -f1 | xargs qdel
For example: User: crlee; Serial number (SN): PQD1;
Then type: qstat -u crlee | grep "PQD1" | cut -d"." -f1 | xargs qdel

**Specific output files:

1. gvcf_list.list: Generated from step 3 (GATK4) or 4 (GATK3). This file will be stored in the "02-get_gvcf_[SN]" folder. This file format also can be used by user defined input sample argument ***--list***.
   This is required for step 3. Please see the details below.
2. [-db folder name]_samples.list: Generated from step 3 (only using GenomicsDBImport). This file will be stored at the same root as the GenomicsDB folder. This file contains a sample list that has been imported into the database, and prevents repeatedly importing the same sample. Please see the details below.
3. c_vcf.list: Generated from step 3 (only using CombineGVCFs) or step 4 (gatk3). This file will be stored at user defined ***-f*** path. By default, it will store at the "03-filtered_vcf" folder. This file contained individual g.vcf path links. This will prevent repeatedly combining the same sample into the final combined g.vcf file (gatk4 with default setting) or import the duplicated samples into final raw.vcf (gatk3).
4. vcf_chr.list: Generated through ***-gv*** argument. This file will be stored at user defined ***-f*** path. By default, it will store at the "03-filtered_vcf" folder. This file contains the path of each filtered vcf file for the gathering step.

Pipeline steps:

Step	**Pipeline function**
0	This step generates an index of the reference file using BWA index function, CreateSequenceDictionary (Picard[GATK3] or Picard[GATK4]) and samtools faidx. The BwaMemIndexImageCreator will also be run (GATK4 only) for following use.
1	Trim raw fastq files. This step has two scripts, including quality check and sequence trimming. Quality check is optional, please see argument ***-q***. For the trimming step, the fastp (or trimmomatic) tool (GATK4) or SolexaQA++ dynamictrim (GATK3) will be used.
2	This step generates raw gvcf files. Basically, the pipeline for this step is: GATK3: bwa mem->samtools view->samtools sort->MarkDuplicates ->samtools index->samtools stats->HaplotypeCaller ->tabix GATK4: (none GRAGEN-GATK) FastqToSam->BwaSpark->MergeBamAlignment->SortSam ->MarkDuplicates->samtools index->samtools stats ->HaplotypeCaller->IndexFeatureFile (GRAGEN-GATK) FastqToSam->BwaSpark->MergeBamAlignment->SortSam

	->MarkDuplicates->samtools index->samtools stats ->**ComposeSTRTableFile->CalibrateDragstrModel**-> HaplotypeCaller->IndexFeatureFile
3	In this step, all the gvcf file inputs will be combined together for each chromosome, so there will be several output gvcf files. Each gvcf file contained one chromosome data of all input samples. **If you have a different batch of samples, the pipeline will automatically combine new samples into the result of old chromosome gvcf files generated by this step. If ***-ow*** is set, the old chromosome gvcf files will be over-write. ** If ***-d*** is set, the pipeline will use GenomicsDBImport function instead of CombineGVCFs function. GenomicsDBImport is very slow. CombineGVCFs is faster. However, CombineGVCFs becomes insufficient when samples are more than 1,500. **If ***gatk3*** is set, this step will be skipped.
4	This step will run GenotypeGVCFs script. However, GATK3 and GATK4 have very different definitions and functions in the script, so the GATK3 pipeline cannot run GenotypeGVCFs in GATK4 toolkits. The GATK4 pipeline also cannot run GenotypeGVCFs in GATK3 toolkits. This step will generate several vcf files. Each vcf file contains one chromosome. If ***-as*** is set, the combined vcf file will contain all non-variant sites. If ***--pre-xlsn (-prx)*** or ***--post-xlsn (-pox)*** are defined, an extra filtering of vcf by SelectVariants will be run.
5	This step filtered vcf from step 4 using perl script. Therefore, **filter_vcf.pl** file should be at the same root of pipeline.pl script or this step will not be executed. If ***-vqsr*** is used, this step and the following steps will be substituted by VQSR pipeline.
6	This step gets bi-allelic SNPs and further filtering. (Step 4 of original pipeline)
7	This step gets the overall depth (summed over all samples) of a site. By default, this step will not be run. If you want to run this step, please use ***-sp 7*** in ***S*** mode or ***-esp 7*** in ***P*** mode. (The last part of original pipeline, step 3)

Arguments:

Required

Working step	**Argument**	**Default value**	**Function explanation**
Except 0s	--path ***-p***	*[]*	This argument is required for the folder path of raw FASTQ data/input data folder.
Except 1s	--reference ***-r***	*[]*	This argument is required for the file path of the reference file (FASTA or gzipped FASTA files are accepted).

Optional

Working step	**Argument**	**Default value**	**Function explanation**
--	--version ***-v***	*[false]*	Show versions of GATK3 and GATK4

All	--step ***-sp***	0p	There are several values including numbers from 0 to 7 plus “***s***” or “***p***”. (0s, 0p, 1s, 1p……7) 7s and 7p is equal to 7. ***s*** mode stands for single step. ***p*** mode stands for pipelined steps. Single step mode will only run the single step script. Pipeline mode will run scripts from user defined start step to the user defined end step (if end step (***-esp***) is not defined, the pipeline will run to step 6).
All	--end-step ***-esp***	6	End step of pipeline. If defined, the pipeline will start from ***-sp*** defined step to ***-esp*** defined step. It takes no function in ***s*** mode.
All	--serial-number ***-sn***	*[]*	In each run, the pipeline will generate a random serial number (SN). Thus, you may not mix different run results. If you want to use the same file from existing SN, you can specify the serial number by this argument, and the pipeline will use the same file of existing SN. If you use ***-f*** to specify the output folder of step 4~7, the serial number will not present on the folder name.
All	--execute ***-exc***	*[false]*	By default, the pipeline will only generate qsub files, but not execute. If set, the script will generate qsub files and execute the qsub automatically.
All	--local ***-lc***	*[false]*	Run the pipeline on local machine. Not recommended. If you do this, it may run for a long long long time.
All	--project ***-proj***	*[]*	Setup the project name for Taiwania 1 Server.
All	--WES ***-wes***	*[false]*	This argument is specifically work with whole exome sequencing data form. When this argument is used, the ***-a*** option will not be applied. ***-ns*** option will also be applied to avoid generating too many files. ***-ip*** will also be applied automatically. ***-l*** argument is required.
1	--quality ***-q***	*[false]*	Quality check of FASTQ file. *SolexaQA++ analysis* function will be used.
1	--trimmomatic ***-trimmo***	*[false]*	When using this argument, Trimmomatic will be used instead of fastp. If your data is from

			NovaSeq sequencing, don't use this argument, because Trimmomatic cannot process polyG issue.
1	--adapter-file ***-adp***	*[]* */[false]*	If ***-adp*** is set, the script will automatically trim the adapters. (adapter sequence file stored at "adapters" folder.) If not, the script will only trim for base quality. **Only work if ***-trimmo*** is used **If set, please indicate the path of the adapter file (GATK4). (See detailed format in *trimmomatic* program) **If using the GATK3 pipeline, you don't need to provide a path of the adapter file, but only accept NEB kit adapter sequence.
2	--group-name ***-g*** (required in ***p*** mode through step 2)	*[]*	The group name is required for step 2. If you run other single steps or pipelined scripts after step 2, this argument is not necessary.
2	--RAD ***-rad***	*[false]*	The RAD mode is suitable for RAD sequence, including DArT data. It will skip the MarkDuplicates step.
2	--dragSTR ***-dstr***	*[false]*	If set, the GRAGEN-GATK pipeline will be used. Two additional steps: ComposeSTRTableFile-> CalibrateDragstrModel will be run before HaplotypeCaller
2	--alternative-pipeline ***-a*** (GATK4 only)	*[false]*	If set, the step 2 of pipeline will be run in an old way: bwa mem->samtools view ->samtools sort instead of FastqToSam->BwaSpark ->MergeBamAlignment.
2	--spark ***-s*** (GATK4 only)	*[false]*	If use ***-s*** argument, HaplotypeCallerSpark will be used instead of HaplotypeCaller in step 2. This is multi-thread programs. However, this is a beta tool in GATK4, so this should not be used for generating data, evaluation only. (Caution: According to GATK4 team, this is a beta version, results might be different)
2	--no-gvcf ***-ng*** (GATK4 only)	*[false]*	If use ***-ng*** argument, HaplotypeCaller step will be ignored. No gvcf file will be generated. If ***-dstr*** argument is also used, it will be ignored too. (Caution: if use this

			argument, the following steps will not be executed properly. Please also use ***-sp*** 2s or ***-esp*** 2 when using this argument.)
2	--ignore-gvcf-number-checking ***-ignc*** (GATK4 only)	*[false]*	If use ***-ignc*** argument, the script will not check if the input sample number matches the generated gvcf file number. (Caution: if use this argument, some samples that failed in step 2 will not appear in the final vcf, and there will be no notification of missing samples.)
2-4	--interval ***-l***	*[false]*	Only map read to the specific regions defined by the interval file. This argument is only worked for ***-wes***.
2-4	--interval-padding ***-ip***	150	Specified the extended regions of the interval region. Only work for ***-wes***. If you don't want to apply the padding region, set it to 0.
3	--use-database ***-d*** (GATK4 only)	*[false]*	This is for step 3. If set, the GenomicsDBImport will be used in step 3. (GenomicsDBImport is really slow, CombineGVCFs is faster. However, GenomicsDBImport can obtain more than 1,500 samples. CombineGVCFs becomes insufficient when samples are over 1,500.) If you have more than 50 samples, this script will do batch import by 50 samples at a time, which prevents qsub running time exceeding walltime limit.
3	--database-path ***-db*** (GATK4 only)	./GenomicsDB	By default, you don't need to use this argument. This is to point out where you want to put your database file in step 3. For detail, please read the GenomicsDBImport function in GATK4.
3	--list (GATK4 only)	02-get_gvcf_[SN]/*.vcf.gz	This is for step 3. If a database sample list (or sample list you want to combine into one file) is provided, the script will determine whether these files are ready for importing. If not indicated, the script will use files in 02-get_gvcf folder to get a list.
3(GATK4),	--no-list-checking	*[false]*	This argument can skip checking imported samples list in

4(GATK3)	***-nlc***		combined gvcfs (GATK4) or raw vcfs (GATK3)(skip checking c_vcf.list). If only certain combined gvcfs (GATK4) or raw vcfs (GATK3) are absent, and you don't want to re-run all the chromosomal/interval files using -ow argurment, you can use this argurment. The pipeline will only re-run the absent combined gvcfs (GATK4) or raw gvcfs (GATK3).
3(GATK4), 4(GATK3), 7, 8	--over-write ***-ow***	*[false]*	For step 3 (GATK4) and step 4 (GATK3): This argument determines whether you want to over-write the existing database/batched g.vcf files or not. If the database exists, the importing method will be set to "***update***" mode instead of "***create***" mode. For detail, please read the GenomicsDBImport function in GATK4. If batched *.g.vcf files exist, this argument will replace the existing files. For step 7 and step 8: This argument will delete existing files generated from previous step 7 and step 8.
3~6	--no-separation ***-ns***	*[false]*	If set, all of the contigs(chromosomes/intervals) will be run in a single file. They will not be separated in step 3 to step 8.
3~6	--prefix ***-pf***	*[]*	Set prefix of each chromosome in the vcf file. If set, only contig with the prefix will be processed. For example, if contig_1 is "Chr01" in the reference file, the prefix will be "Chr". If it is "ch01", the prefix will be "ch". The prefix is case-insensitive. If you want to filter for more than one prefix, you can use "1st_prefix|\^2nd_prefix|\^3rd_prefix…". Example: -pf chr|\^mito|\^chloro (it will select contig name start with "chr", "mito", or "chloro".
3~7	--folder ***-f***	./03-filter_vcf_[SN]	This is for step 4~7. By default, the chromosome separated raw vcf files, the

			combined raw vcf file and the filtered vcf file will be stored at ./03-filtered_vcf_[SN] folder. If you want to change the path, you can use this argument to redirect the path.
4	--all-sites ***-as***	*[false]*	This is used for step 4 GenotypeGVCFs function. In GATK3, it equals the “--includeNonVariantSites” argument. In GATK4, it equals the “-all-sites” argument. This argument also applies to SelectVariants function.
4	--pre-xlsn ***-prx*** (GATK4 only)	*[]*	If set, please indicate the file path of sample list end with extension “.args”. When using this argument, SelectVariants function will turn on. The list is samples you want to **exclude** from the vcf file. One sample name per line in the list file. This function acts before GenotypeGVCFs. If you want to use more options in SelectVariants, please use **SelectVariants_qsub.pl** to send job(s) or use original SelectVariants function in GATK4. **This function works prior to -***prn***
4	--pre-sn ***-prn*** (GATK4 only)	*[]*	If set, please indicate the file path of sample list end with extension “.args”. When using this argument, SelectVariants function will turn on. The list is samples you want to **include** from the vcf file. One sample name per line in the list file. This function acts before GenotypeGVCFs. If you want to use more options in SelectVariants, please use **SelectVariants_qsub.pl** to send job(s) or use original SelectVariants function in GATK4.
4	--post-xlsn ***-pox*** (GATK4 only)	*[]*	This is similar to --pre-xlsn, but doing SelectVariants after GenotypeGVCFs. **This function works prior to -***pon***
4	--post-sn ***-pon*** (GATK4 only)	*[]*	This is similar to --pre-sn, but doing SelectVariants after GenotypeGVCFs.

4	--pre-selected ***-ps*** (GATK4 only)	*[]*	Path of pre_select gvcf files stored. (Should direct to folder path not file path.)
4~6	--gather-vcfs ***-gv*** --gather-vcfs-cloud ***-gvc***	*[]*	If set, the gathering step will be run. All of the contig (chromosome/interval) vcf files will be gathered as a single vcf file. ***-gvc*** uses multi-thread mode in the step.
5	--VQSR ***-vqsr***	*[false]*	Using VQSR mode. This argument will substitute the step 5 to 7. The ***-res*** argument is required.
5	--resource ***-res***	*[]*	The source of VQSR filtering reference. Only work for ***-vqsr***. Please see the format at resources_example.txt.
5	--assembly-name ***-an***	GRCh38	This argument is used in VEP step of VQSR mode.
5~7	--skip-filtering ***-sf***	*[false]*	If set, step 5 will be skipped.
6	--bi-allele-off ***-bao***	*[false]*	By default, vcftools will use "--min-alleles 2 --max-alleles 2" for bi-alleles selection. If you don't want to select bi-alleles, use ***-bao*** to turn off this option.
6	--with-indels ***-wi***	*[false]*	By default, vcftools will remove indels. If you want to retain indels, use this argument.
6	--max-missing-count ***-mmc***	*[]*	Filtering missing site by missing numbers of total sample numbers. Possible value should be an integer. If set, this argument is prior to ***-mm***.
6	--max-missing ***-mm***	0.9	Filtering missing rate by total samples. If set to 0.9, it means that only retains sites with missing rate less than 10%. Possible value should be between 0~1. Set to 0 if you don't want to filter by this option.
6	--maf ***-maf***	*[]*	Filtering MAF of the alleles. Possible value should be between 0~1. This filtering is off by default.
6	--minQ ***-mq***	30	Filtering variant sites by base quality. Possible value should be an integer. Set to 0 if you don't want to filter by this option.
6	--keep ***-kp***	*[false]*	Filtering to keep only user defined samples as the function in vcftools --keep.
7	--prefix-DP ***-pfdp***	chr	Only grep vcf variant have define prefix in CHROM field.

			For example, if CHROM is "Chr01" in the vcf file, the prefix will be "Chr". If it is "ch01", the prefix will be "ch". The prefix is case-insensitive.

Environment setup:

```
# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/softwares/vcftools_0.1.13/bin:$HOME/softwares/bwa:$HOME/softwares/htslib:$HOME/softwares/samtools-1.4.1:$HOME/softwares/bcftools:$HOME/softwares/SolexaQA_v3.1.7.1:$HOME/softwares/plink_linux_i686_1.9:$HOME/softwares/sratoolkit.2.8.2-1-ubuntu64/bin:$HOME/softwares/sshpass-1.06

# It is better to add following lines to ~/.bashrc
export PATH
export PERL5LIB=$HOME/softwares/vcftools_0.1.13/perl/
export BCFTOOLS_PLUGINS=$HOME/softwares/bcftools/plugins/
export TRIMMO=$HOME/softwares/trimmomatic.jar
export LC_CTYPE="en_US.UTF-8"

GATK3:
export GATKFILE=$HOME/softwares/GATK_3.7/GenomeAnalysisTK.jar
export PICARDFILE=$HOME/softwares/picard_2.9.0.jar

GATK4: (Please check java version, OpenJDK 1.8 or Java 8 is required)
export gatk2=$HOME/softwares/GATK_4.2.2/gatk
alias gatk2='$HOME/softwares/GATK_4.2.2/gatk'
```

Additional tools for the pipeline:

1. **qstat_check.pl**

   Usage: perl qstat_check.pl [SN]

   SN=Serial number, it is optional.
   This script will detect how many jobs are still running. If SN is provided, only the job name with SN will be detected. Otherwise, all of the GATK jobs will be detected.

2. **downsampling.pl**

   ```
   Usage: perl downsampling.pl -p [0~1] -i SAMPLE.fastq
                 or
          perl downsampling.pl -p [0~1] -i SAMPLE_R1.fastq
          SAMPLE_R2.fastq
   ```

   This script can downsampling a large fastq file (single-end or paired-end, *.gz file is acceptable). -p value is between 0 to 1. It means the percentage of subset to be downsampling. -i is input file path. Paired-end files should indicate as two file paths in -i argument. This script also automatically compresses unzipped file(s) to save storage space.

3. **SelectVariants_qsub.pl**

Usage: perl SelectVariants_qsub.pl -R REFERENCE.fasta -V INPUT_FILE -O OUTPUT_FILE [--select-type-to-include SNP][-xl-sn LIST.args][-sn LIST.args][--exclude-non-variants][-L INTERVALS]

This script will filter vcf file as user defined. Which is basically the same with SelectVariants function in GATK4 with additional qsub command lines. However, only limited arguments are supported.

4. create_job.pl

Usage: perl create_job.pl COMMAND_LINE [-cj_exc][-cj_sn][-cj_help]

This tool is for generating qsub command line file. You can just put any shell command line, and it will generate a qsub file. For special characters used by shell (eg. “>”,“|”), please add “\” in front of the characters. If you want to send the job immediately, please add “-cj_exc” in the command line. -cj_sn: this tool will generate a 4-digit random serial number (SN). If you want to use previous generated SN, you can use this argument to define SN. By default, it will use 1 node, 2 ppn and 16Gb memory. If you want to change the setting, please modify the perl file at line 6, 7, and 8. If mem set to 0, it means memory is not defined.

5. count_vcf_missing.pl

Usage: count_vcf_missing.pl VCF_FILE

This tool will export missing SNP number and rate, and also heterozygote allele number and rate. The exported file will be “statistics_vcf.txt”. gz file is acceptable.

6. filter_vcf_2.4.pl

Usage: perl filter_vcf_2.4.pl INPUT_RAW_VCF

This is a bundle script of GATK_pipline.pl, which is required for filtering step. Basically, it is the same with 03-Filter_vcf.R file, only difference is this is a perl version. Also, it fixed a bug that when two alleles are separated by “|” but not “/”, R script will be shot down. No output path needs to be indicated.