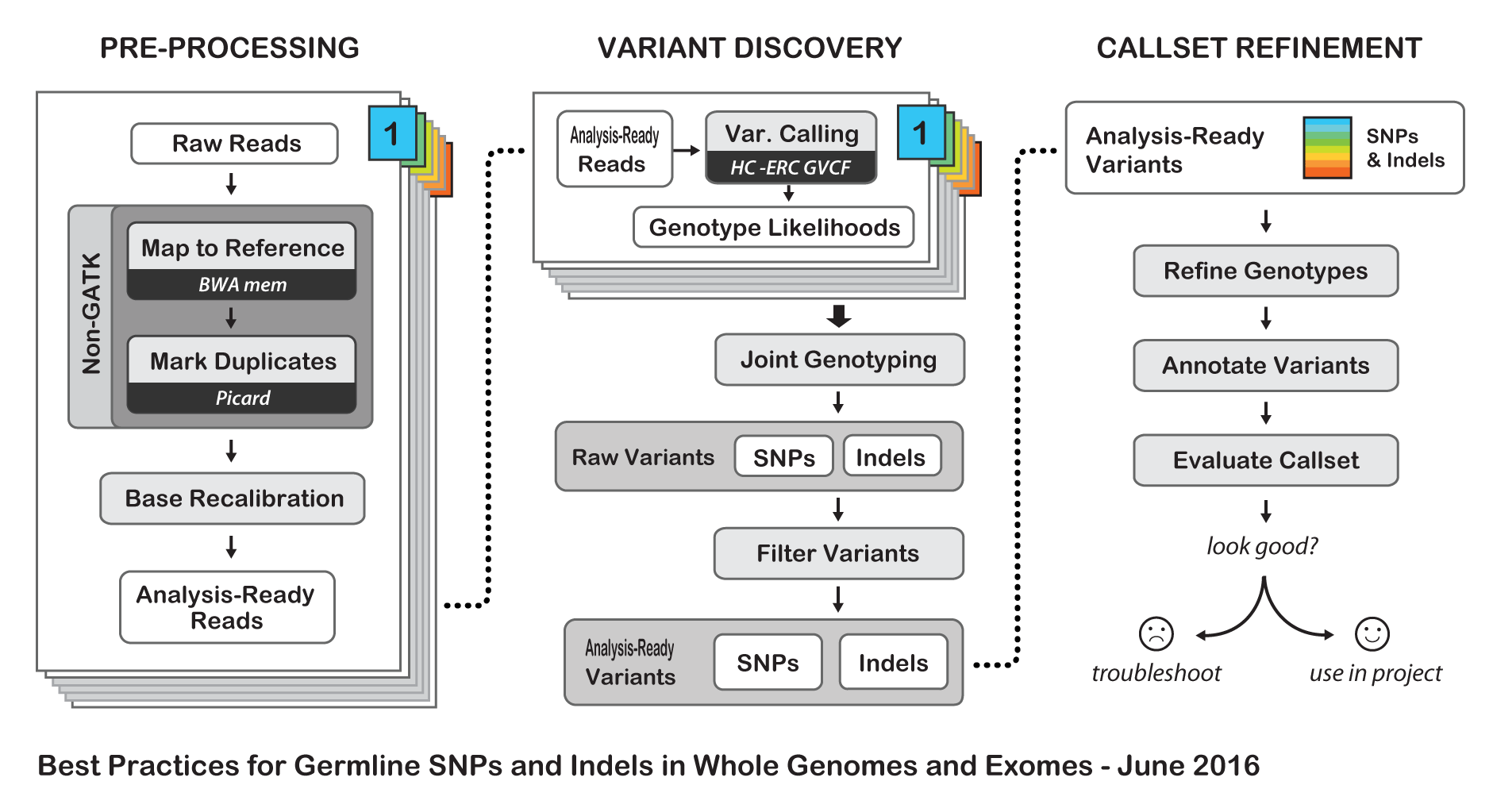
User Documentation 

# Pipeline architecture and function

This pipeline implements the GATK's best practices for germline variant calling in Whole Genome and Whole Exome Next Generation Sequencing datasets (<https://software.broadinstitute.org/gatk/best-practices/>), given a cohort of samples.

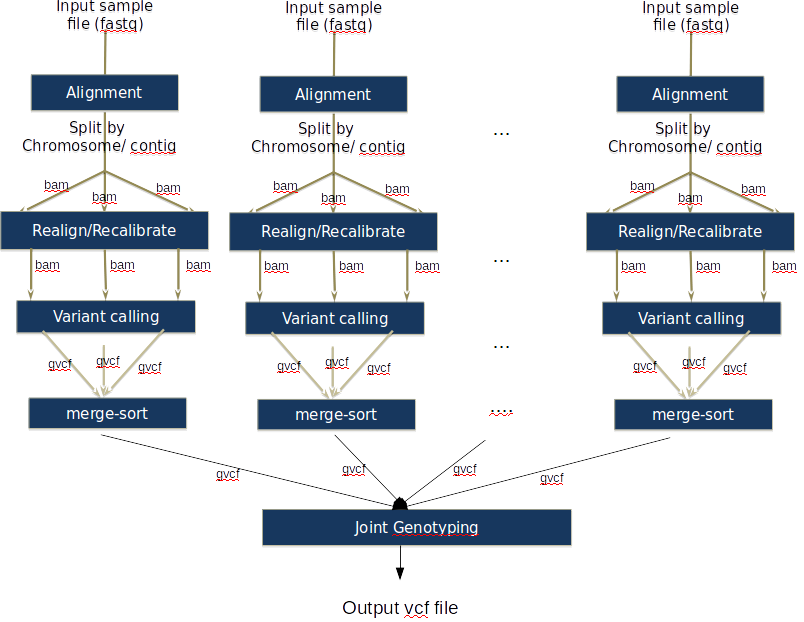
In its latest version, 3.6, the best practices include the stages shown in Figure [1] below, which are:

1. Mapping to the reference genome
2. Marking duplicates
3. Base recalibration (BQSR)
4. Variant calling –----- (processing done per sample)
5. Joint genotyping –----- (processing done for all samples together)

These stages are implemented in this pipeline, with an optional “Indel Realignment” step (which was required in previous GATK best practices < 3.6)

The pipeline can be run as: Alignment stage only, Complete variant calling with realignment and Complete variant calling without realignment

Practically, running the pipeline requires parallezation of some of the stages, which is achieved as in Figure [2] below:



# Dependencies

The pipeline implements the stages of Figure [1] and [2], while allowing different software tools at some of the stages depending on user's preference. These are as follows:

|  |  |
| --- | --- |
| Stage | Tools |
| Quality control | Fastqc: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/ |
| Illumina reads trimming | Trimmomatic: http://www.usadellab.org/cms/?page=trimmomatic |
| Alignment | Bwa mem (<https://github.com/lh3/bwa>), Novoalign (<http://novocraft.com/> ) |
| Marking duplicates | Samblaster (<https://github.com/GregoryFaust/samblaster>), Novosort (<http://novocraft.com/> ), Picard (<https://broadinstitute.github.io/picard/>), |
| Indel realignment | GATK () |
| Base recalibration | GATK () |
| Calling variants | GATK (Haplotypecaller: https://www.broadinstitute.org/gatk/gatkdocs/org\_broadinstitute\_gatk\_tools\_walkers\_haplotypecaller\_HaplotypeCaller.php) |
| Jointcalling | GATK (Genotypegvcf: https://www.broadinstitute.org/gatk/gatkdocs/org\_broadinstitute\_gatk\_tools\_walkers\_variantutils\_GenotypeGVCFs.php ) |
| Miscelleneos | Samtools (http://samtools.github.io/) |

## Repo index/ The scripts

The files in the repo: https://github.com/HPCBio/BW\_VariantCalling/tree/ParameterSweep

\* please proceed as follows:

- start.sh: the main file to start the workflow

- align\_dedup.sh: the script to the alignment job

- realign\_varcall\_by\_chr.sh : the script to the realignment and variant calling jobs

- merge\_vcf.sh: the script to merge the bams (we split the per-sample bams by chromosome in the realignment /recalibration stage, and now we merge that). Similarily, we merge the vcfs

- joint\_vcf.sh: the script to do joint variant calling

- summary.sh: the script to generate summary about the last run of the pipeline (which jobs/samples failed/succeeded)

\* Unused scripts (in the current form of the pipeline):

- trim\_input.sh: a script to do quality control: run fastqc then trimmomatic to trim adapter sequences from reads

- start\_vqsr.sh: A script (pipeline to do VQSR) to the per-sample raw vcf file after calling the variants using the start.sh pipeline. It does so by calling the recalibrate\_vcf.sh script. It expects a sampleinfo file that ...........

- recalibrate\_vcf.sh: a script to do VQSR to both snps and indels. Actually, in its current form it doesn't produce the right result (snps should be vqsr'd in both steps before indels. it wouldn't work when both models are built first and then applied)

- mutect2\_1pair.sh: script to call variants in normal/tumor samples using mutect2 from the GATK

- read\_fate.sh: script to generate an overall summary of the alignment job (parsed from the alignment log files) (total number of reads, total mapped reads, total duplicates, percentagae duplicates, percentage mapped)

- index\_bams.sh: a script to index recalibrated bam files. It is said to be needed to run Mutect2, but not sure

- merge\_bam.sh: a script to only merge bam files. Not really needed, as that is done in the merge\_vcf.sh file (it was developed for a different purpose, and later found unnecessary)

- varcall\_by\_sample.sh: this was done when the believe was that variant calling should be done per sample. This was found to not be necessary!

- resumePartialExectuion.sh: script to resume the execution of the start.sh workflow in case of partial success of some of the steps

- vcfstats.sh: script to generate vcf stats on all vcfs resulting from the workflow (QC step)

## Runfile options

## Samplenames file format

## Step-by-step instructions for setting up the runfile

# Changelog

Changelog 2.32

* samplenames are no longer listed in runfile
* epilogue is gone
* new convention on input fastq names: must be in form of samplename\_read?.fq or samplename\_read?.fastq
* runfile now must contain INPUTFORMAT flag to dostinguish between fastq and bam

Changelog 2.33

* alignfastq.sh -- changed names of all qsub variables to be more descriptive
* realign.sh -- changed names of all qsub variables to be more descriptive
* realign\_new.sh -- changed names of all qsub variables to be more descriptive
* alignfastq.sh -- realign.sh is now submitted with a dependency on merge jobs
* realign.sh -- now runs
* now in runfile the chromosomes are specified explicitly as given in the full
* genome fastq, not just the numbers; this way any species can be analyzed;
* made appropriate changes in realign\_new.sh
* introduce the Anisimov switch: \*\* want to run real/recal and variant calling via the Launcher. But, realign\_new still has to be a separate scheduled job, because there is a case when it depends on sortbammayo and extract\_reads\_bam: when bams were obtained from elsewhere and need to be resorted. \*\* introduced user-level variable RUNMETHOD ($run\_method), with values: \*\*\* ANISIMOV - means we use the anisimov launcher \*\*\* QSUB - means we schedule each job as an individual qsub without aprun \*\*\* APRUN - means we schedule each job as an individual qsub with aprun \*\*\* SERVER - means we run the entire workflow on the same machine in series
* renamed user-define variable TYPE (input\_type), because "type" seems to be some sort of a special variable (highlighted in the editor), even though it is not a reserved variable in bash...
* introduced variable "profiling" to choose whether to use memprof, cray Profiler or something else
* realign.sh calls realign\_new.sh; in 2.32 it calls one realign\_new per sample in the case of independent samples, and a single realign\_new for the multisample case. In either case, one of the input parameters is the directory where to look for the aligned bams. I am rewriting the code so that a single realign\_new is called in any case, with the appropriate folder for aligned bams, depending on the case.
* moved creating SAMPLENAMES.list file into main.sh -- should not have to do this in every piece of code!!
* intended to run all realrecall in a single joblist per chromosome. But, multisample case makes it more sensible to have one joblist for sortnode, as it is independent of multisample variable; then if multisample then we only use a single qsub for realrecalold, and a single one for vcall; but for independent samples, we do one joblist for realrecal per chromosome, and one joblist for vcall per chromosome
* for each qsub file, we now construct a command file. because we have extra flexibility of using qsub, aprun, launcher or serial; then the command gets modified and contents put into qsub or jobfile

NEED TO RENAME SORTNODE AND REALIGN, REALIGN\_NEW AND REALRECAL.OLD

also vcallgatk, vcallmain mainaln

# Test datasets

# Table of Contents

* Pipeline architecture and function
* Runfile options
  + i/o
  + choose the run case
  + input data
  + tools to be used
  + preparatory block
  + alignment block parameters
  + realign/recalibrate block parameters
  + variant calling block parameters
  + other parameters - DO NOT EDIT
  + paths to input output and tools - DO NOT EDIT
  + pbs resources - DO NOT EDIT
* Samplenames file format
  + Single bam input
  + Single fastq input
  + Multiple bam inputs
  + Multiple fastq inputs
* Step-by-step instructions for setting up the runfile
  + Case 1: only alignment
  + Case 2: only realignment /recalibration
  + Case 3: only variant calling
  + Case 4: alignment + realignment /recalibration
  + Case 5: realignment /recalibration + variant calling
  + Case 6: entire pipeline = alignment + realignment /recalibration + variant calling.