**GeneMAP NGS Variant Calling Pipeline for Paired End Reads**

# Introduction

Considering the large genomic data sets that are now being generated from GeneMAP projects and across study sites, a unified pipeline for handling raw sequence data as they emerge from the sequencing platform to analysis-ready BAM or GVCF files is required. This will facilitate among site data analysis comparisons, in addition to facilitating the timely processing of data at the production level. Hence the goal of this pipeline is to automate the processing of raw sequences from Fastq to analysis ready BAM/GVCF for GeneMAP project sites.

# What will the Pipeline accomplish?

The pipeline will handle the processing of raw sequence files from Fastq quality check, trimming of poor quality reads when necessary, mapping/alignment, variant calling and variant filtering, and annotation. The pipeline is motivated by the GATK best practices. Thus, the steps from mapping/alignment and downstream depend heavily on GATKv4.x. While the pipeline seeks to automate the variant calling pipeline from fastq to BAM/GCVF, we are aware that a user may choose to enter the pipeline at any stage given what data is available. For instance, the user may already have BAM files to only do variant calling and downstream analysis. Or the user may have GVCF files. These can all be handled by simply providing a list containing the BAM/GCVF files and the path to the files.

Although variant calling by BCFTOOLS is currently added, it may not be required while GATK is installed.

# What has been added thus far?

## Fastq Quality Check (FastQC)

Initial quality check of raw Fastq/SAM/BAM files prior to any further analysis. This step is highly recommended when for any analysis for which there is no prior information about its processing or its processing history.

* Program: Fastqc
* How was it implemented?

This step only requires the user to specify the path to the Fastq/SAM/BAM files, making sure that ONLY THE SEQUENCES TO BE PROCESSED ARE FOUND IN THE PATH. Also, the user may decide to run this step in parallel, in which case the user will ensure that they have GNU parallel installed.

## Trimming of poor quality reads

This will be informed by the Fastqc step. Poor reads will be trimmed as necessary.

* Program: Trimmomatic
* How was it implemented?

Again, this step will require the user to specify the path to the Fastq files. Only the files to be trimmed MUST be present in the path. The user may also specify an adapter, in which case the adapter should be present in the current working directory (directory from where the geneMapNGS script is run). The number of bases to trim from both 5’ and 3’ end may be specified as well, and again, the user may choose to run the job in parallel (i.e. more than one sequences may be trimmed at the same time).

## Mapping/Alignment

This step only takes Fastq files as input (for now). If realignment of BAM files is required, they should be converted to Fastq files before running this step (BAM to Fastq file conversion commands may be added according to the need).

When running all the steps required from fastq to BAM/GVCF, the script will prioritize GATK for variant calling. Here, when the user is running the entire pipeline at once, the script will create directories for each main output, and will look for input to the next step from the previous directory it created. This will be the default behavior of the script in accordance with the goal of the pipeline; which from Fastq-to-BAM/GVCF. The user will also be required to provide a sample/metadata file containing four columns as described below;

Column 1: Forward Fastq reads

Column 2: Reverse Fastq reads

Column 3: Sample name

Column 4: Sequencing platform

### Example:

|  |
| --- |
| XXX\_L001\_R1\_001.fastq.gz XXX\_L001\_R2\_001.fastq.gz SAM1 ILLUMINA |
|  |

ERRXXXXXXXXX\_1.fastq.gz ERRXXXXXXXXX\_2.fastq.gz SAM2 ILLUMINA

### Why the sample/metadata file?

A recurrent problem with alignment/mapping using BCFTOOLS is specifying the sample name, read group, platform name, and other useful information that are needed in the header. Although BCFTOOLS has an argument to take of this issue, the implementation requires for a rather more interactive job processing, or some level of knowledge of command line syntax in order make this work in an automated fashion. In response to this challenge, the GATK team developed a simple model that requires a sample/metadata file and an unmapped BAM file (which can be readily obtained from Fastq files using GATK’s ***FastqToSam*** command). Once variant calling is performed with GATK’s ***HaplotypeCaller*** and mapped BAM files are obtained, GATK uses its ***MergeBamAlignment*** command to consolidate the information contained in the unmapped BAM files with the mapped BAM files, with some level of quality check to produce mapped BAM files that contain all the information in the header needed for downstream analysis. An important feature of the pipeline is that it will take a pedigree file (optionally). This file is used to determine the population founders, and thus used to compute population structure annotations like inbreeding coefficients which will be later used to assess variant qualities by variant quality score recalibration (VQSR). It should be noted that the pedigree file is NOT used in the actual variant calling process (GATK Team). If provided, the PED file should have the following information;

Column 1: Family ID

Column 2: Individual ID

Column 3: Paternal ID

Column 4: Maternal ID

Column 5: Sex: (1=male, 2=female, other=unknown)

Column 6: Phenotype (1=unaffected, 2=affected, 0=missing, -9=missing)

### Example:

FAM001 FAM001IND1 FAM001P1 FAM001P2 1 2

FAM001 FAM001IND2 FAM001P1 FAM001P2 1 2

FAM002 FAM002IND1 FAM002P1 FAM002P2 1 2

FAM003 FAM003IND2 FAM003P1 FAM003P2 1 2

**NB**: If the user already has BAM or GVCF files, then they will be required to provide a file containing the BAM/GVCF files, and specify the path to the files. For BAM files, the user must ensure that the read group information (which is usually the same as sample name) in the header is provided, else GATK will error out.

NB: Mapping/Alignment is performed using BWA-MEM

# Variant Calling by GATK: Why GATK?

GATK provides a suit of commands that are powerful for handling NGS/Exome data. Enhanced features for reads mapping/alignment optimization (like indel realignment and base quality score recalibration) and variant quality assessment and filtering. If GATKv4.x is used, then the step of indel realignment that is usually required after alignment with BWA-MEM prior to variant calling by BCFTOOLS will not be required since GATKv4.x’s *HaplotypeCaller* command performs some local realignment. Hence, the default behavior of the script is to not perform this step.

## How does it flow?

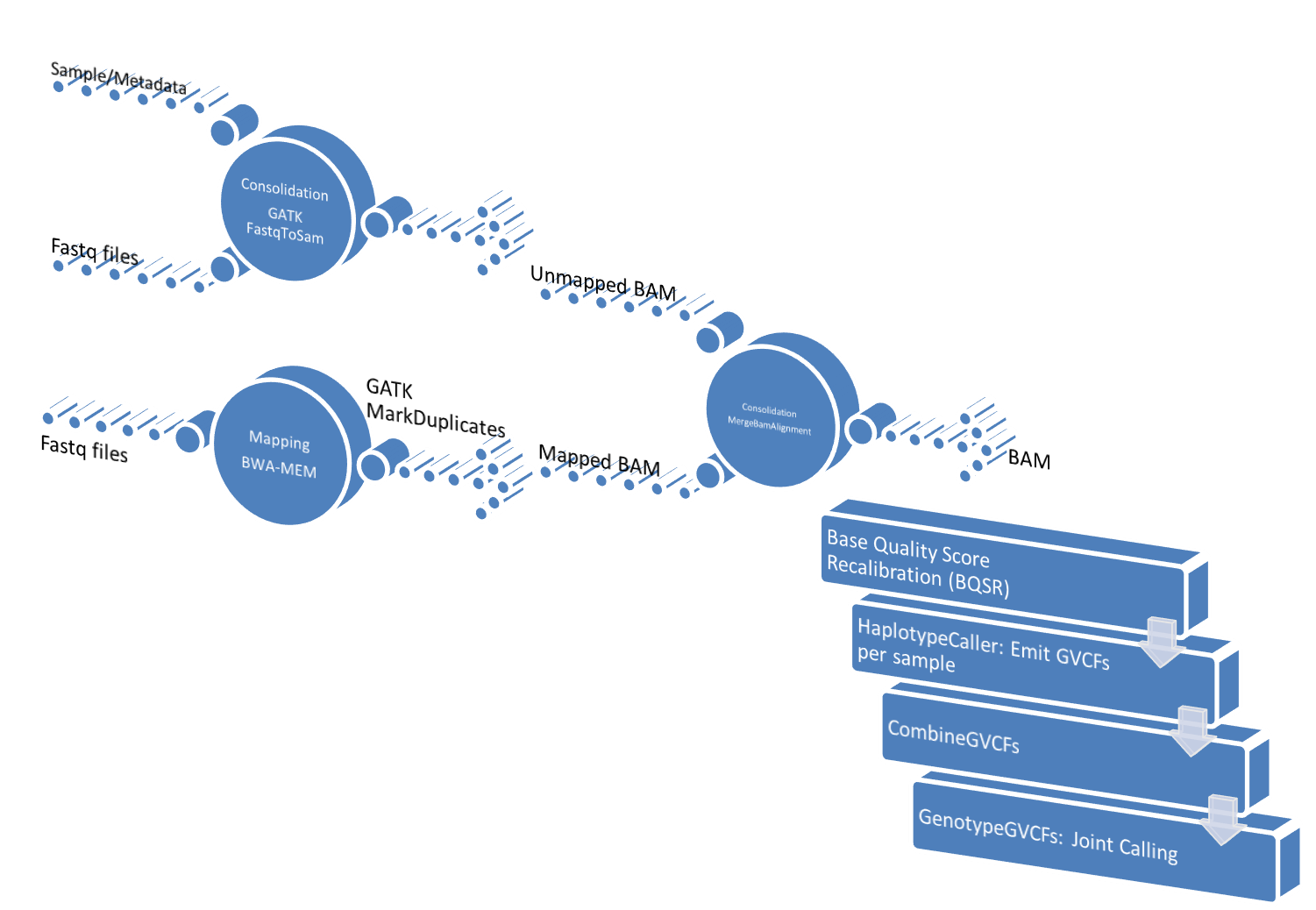


Figure 1. Variant Calling Pipeline

## Single calling, batch calling or joint calling?

* **Single calling**: refers to per sample variant calling. This is not recommended due to the low power that comes with calling variants in a single sample. Variants may not be called at positions without sufficient read depth.
* **Batch calling**: Sometimes the sequencing machine emits sequences in batches. This is especially true when large sample sets are being sequenced. Usually, the samples are processed as they are emitted. This would be considered batch processing. While the samples may be of sufficient size to power variant calling, the systematic differences that may arise from processing different batches like a batch of samples showing marked higher mapping quality than others, may pose a challenge for downstream analysis. Family-wise joint calling can be classified under this category.
* **Joint calling**: Involves variant calling on all the samples. This is recommended as it sufficiently powers variant calling. The increase in read depth means sensitivity for variants with low read depth is increased.

## Variant Filtering

After joint calling, it is important to assess the quality of called variants, and discard false positives and poor quality variants. Two approaches are commonly used to achieve this; hard-filtering and variant quality score recalibration (VQSR)

### Hard-filtering

This involves the use of various filtering annotations like mapping quality (MQ), Phred scaled score by the fisher exact test (FS), inbreeding coefficient, etc.

### VQSR

This is a machine learning technique that utilizes the information obtained from known variant files and specific annotations to model false positive variants estimate variant qualities. This is a preferred procedure and has been added to the pipeline.

The GATK team has great information on this (<https://gatk.broadinstitute.org/hc/en-us/articles/360037499012-I-am-unable-to-use-VQSR-recalibration-to-filter-variants>)

# What is left to be done?

While this script does the job, it is not in the preferred language for NGS bioinformatics. It is a combination of pieces of bash codes that have accumulated following NGS/Exome analysis. The entire pipeline will be implemented in python (which is the preferred language) in due course.

Scripts for ANNOVAR annotation have not been added to this pipeline owing to the fast that ANNOVAR would require several databases with a dedicated path. Thus the following are recommended for data analysis for the various GeneMAP sites;

* Capture pedigree information for family-based analysis according to the PED file specification above. Ensure sample names are labeled according to a generally accepted guideline (standard) across sites/users.
* Capture raw sequence data information across sites/users in a unified manner according to the sample/metadata file specification above.

These recommendations are motivated by the necessity to standardized the pipeline across sites/users such that a common repository containing all databases (human reference sequence, dbSNP, ANNOVAR annotation databases, hapmap SNPs and other known variants databases, etc) will be created and pointed to the script at each site’s/user’s installation of the pipeline. This would greatly facilitate running of the pipeline without much interaction with it. Therefore, the final version of such script would contain a setup script that will create a uniform environment for the pipeline across sites/users.

# Reference

McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., … DePristo, M. A. (2010). The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. Genome Research. https://doi.org/10.1101/gr.107524.110