Replicate GATK Variant calling analysis

Gialitsis Nikolaos DS2.190005

1. Brief Description of the analyzed data and overview

My task was to perform an analysis on 15 discrete Human samples (HG00234-HG00251) in order to identify variants, a process referred to as “Variant Calling” using the recommended protocol from the Broad Institute.

https://gatk.broadinstitute.org/hc/en-us/articles/360035535932-Germline-short-variant-discovery-SNPs-Indels-

Only exome alignments were utilized, so in this case we are interested in variants in protein coding regions, which is practical since a drug discovery process usually transpires around targeting a single problematic protein, so identifying SNPs is an important step towards Personalized Medicine.

Moreover, the reads result from germline sequencing, so we are interested in inherited variants, in contrast to somatic variants which are local and dynamic in nature. One important thing to note, is that all germline cells in a single body carry the same genetic information whereas somatic cells are different depending on their location , for example cancer is the result from mutations on a subset of all cells.

The reads should are resulting from sequencing the chromosome 11. The chr11 contains about 135M bps and represents 4-4.5% of total DNA in cells. It also contains 1300-1400 protein coding genes.(https://ghr.nlm.nih.gov/chromosome/11) . However for the first steps of the downstream analysis, I downloaded the whole reference genome and not just the particular chromosome.

Also, the reads are paired-end, so this results in more accuracy. I run the following command to investigate if they are single-ended or pair-ended:
samtools view -c -f 1 file.bam
an output of ‘1’ means that the reads are paired-end (as in our case).

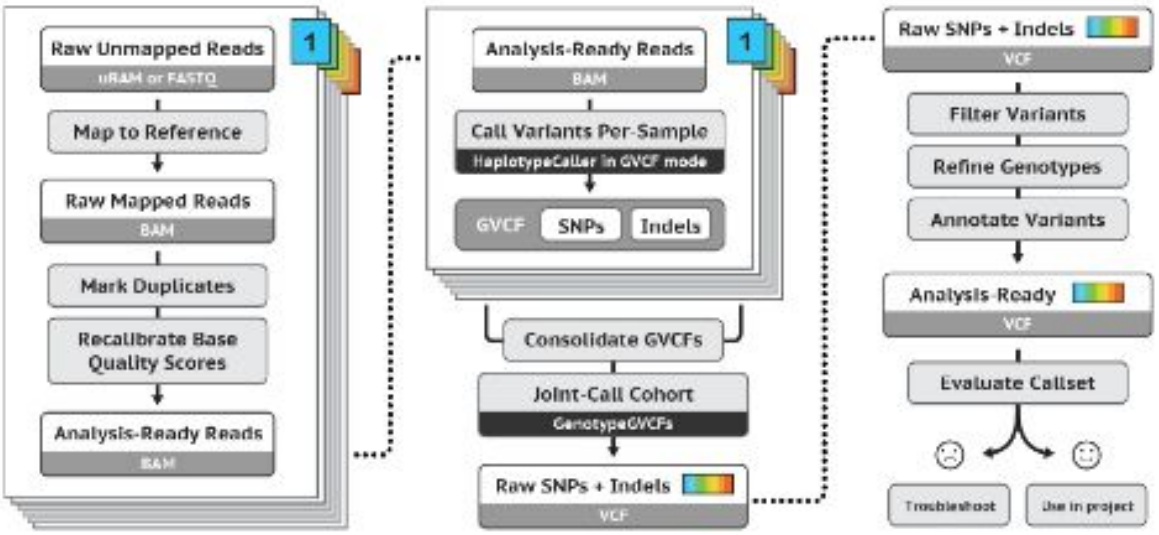
While discussing with my colleagues with a biological background, I expressed the question, whether a single-nucleotide variant is the same as a mutation. This caused some debate but we found the following chart explaining the differences:
(https://biodifferences.com/difference-between-mutation-and-variation.html)

BASIS FOR COMPARISON	MUTATION	VARIATION
Meaning	Mutation is the natural and permanent change, causing changes in the DNA sequence in any living organisms.	Variation or genetic variation is seen in an individual of any species, groups or population and is observed in genes as well as in alleles. By the process of natural selections, mutations may bring evolutionary changes.
It affects	Mutations affect the single organisms.	Variations are seen in groups or populations of an individual.
Causing agent	Chemicals, ionising radiations, radioactive rays, chemical mutagens or x-rays.	Gene mutations, crossing over, genetic recombinations, genetic drift, gene flow, environmental factors.
Types	1.Germline or Hereditary or Fixed or Stable or Chromosomal mutations. 2.Somatic or Acquired or Dynamic or Unstable mutation.	1.Environmental variation. 2.Genetic variations. 3.Continuous variation. 4.Discontinuous variation.

So one important difference is that when talking about mutations, we refer to a change in a single individual whereas a mutation refers to genomic changes in groups of people, so the emphasis is on the population.

2. Protocol all analysis steps

The Broad Institute has developed the GATK pipeline which is summarized in the following steps:



Preprocessing and mapping reads to reference

The input consists of the set of unmapped reads in raw form

Next, we map the reads to a reference genome, in our case the b37 human genome, also referred to as the hg19 reference
https://gatk.broadinstitute.org/hc/en-us/articles/360035890711-GRCh37-hg19-b37-humanG1Kv37-Human-Reference-Discrepancies

To index the reference genome and map the reads to it, I used the bwa aligner aka. the Burrows-Wheeler Aligner, which is recommended by the best practices protocol. bwa utilizes the Burrows-Wheeler Transform, a data compression method useful for string containing runs of characters such as in DNA:

Transformation				
1. Input	2. All rotations	3. Sort into lexical order	4. Take the last column	5. Output
^BANANA|	^BANANA| |^BANANA A|^BANAN NA|^BANA ANA|^BAN NANA|^BA ANANA|^B BANANA|^	ANANA|^B ANA|^BAN A|^BANAN BANANA|^ NANA|^BA NA|^BANA ^BANANA| |^BANANA	ANANA|^B ANA|^BAN A|^BANAN BANANA|^ NANA|^BA NA|^BANA ^BANANA| |^BANANA	BNN^AA|A

The **bamToFastq** command was run in order to separate the paired-reads into fastq files and then map them to the reference. The fastq files report the mapper quality using quality scores and cigar strings.

The **bwa index** command was used to index the reference h19 genome and the **bwa mem** command was used to map the reads to sam.

Of course, before performing indexing, I sorted the bam files using **samtools sort**.

Marking Duplicates and Recalibration

Duplicates (optical or library duplicates) affect the final variant call as the duplicated read can be interpreted as evidence towards a certain SNP, which is something that we have to avoid. The more you sequence a library the higher the number of duplicates. The number of optical duplications (nearby clusters on flow cell) is consistent because it is a preparation error.

In order to find duplicates, the algorithm does not examine the sequences themselves because sequences contain errors. The focus is on the positions of mapped reads (e.g where were the first reads mapped to) . So if two reads have their first mapped read in the same position, they are probably duplicates. A common practice is to identify duplicate sets and choose the most “representative”.

A duplicate status is indicated in SAM. Duplicates are usually not removed but tagged with a corresponding flag. Downstream analysis usually ignores the flagged reads and most GATK tools ignore those by default.

There are only a few cases where we might want to keep duplicates, for example in Amplicon Sequencing where all reads start at the same position by design. Another example would be when conducting allele-specific expression analysis.

The problem of marking duplicates is NP-hard

In order to mark the duplicated reads, I used the Picard tool, also developed by the Broad Institute and specifically I run the following command (all commands are in the log file submitted with this report)

java -Xmx2g -jar MarkDuplicates.jar INPUT=$file.sorted.mapped.bam OUTPUT=$file.dedup_reads.bam METRICS_file=$file.metrics.txt

A thing that I would like to mention is that I worked on the okeanos virtual machine, which had the java version 1.7 installed but the latest support for picard requires jdk 1.8. At first, I thought that updating the java version would be more troublesome than finding an older version of picard, I used the picard version 1.119 which support jdk 1.7. It turns out this was not a smart move since I was forced to update the java version in order to use gatk later on.

This process resulted in new files which (hopefully) do not contain any duplicates. I had to sort and index those bam files as before and deleted the previous “dirty” reads.

Next, I used the picard AddOrReplaceGroups command on the remaining files in order to fix any errors related to the read groups after the duplicate elimination.

Next we have to perform Base Quality Score Recalibration (BQSR) corrects the machine error after removing the duplicates.

for this step, I used **gatk BaseRecalibrator** with the following command:

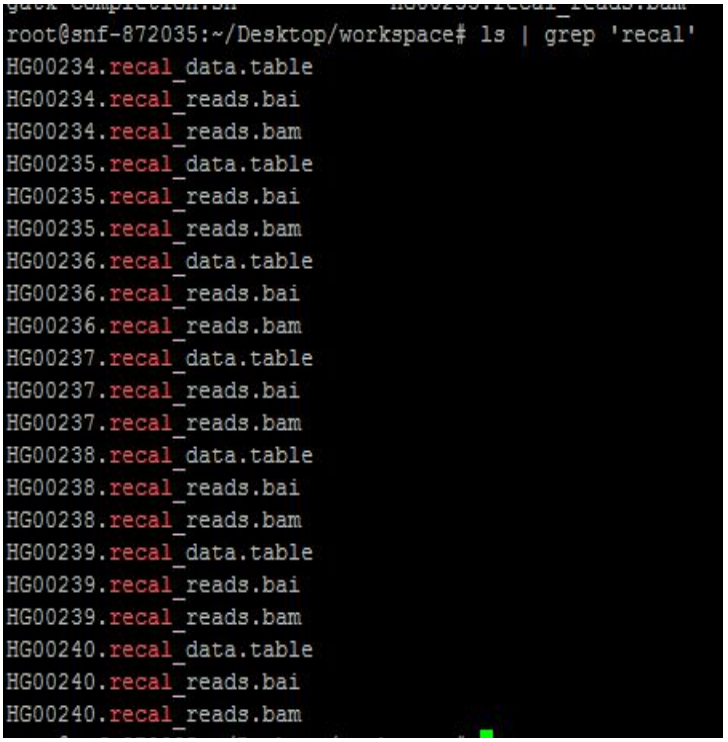
```
gatk BaseRecalibrator \
-R b37.fasta \
-I $file.bam \
-L 11 \
--known-sites dbsnp_138.b37.vcf \
--known-sites Mills_and_1000G_gold_standard.indels.b37.vcf \
-O $file.recal_data.table
```

where I have used two datasets containing known variants . To do that, I run the **wget -c** and **gunzip -d** commands to store the data in .vcf format.

the -L 11 argument means that we are interested only in reads mapped to the chromosome 11.

In addition, the gatk recalibration step required a dictionary for the reference. This was done with the picard's CreateSequenceDictionary command which creates the b37.dict output file.

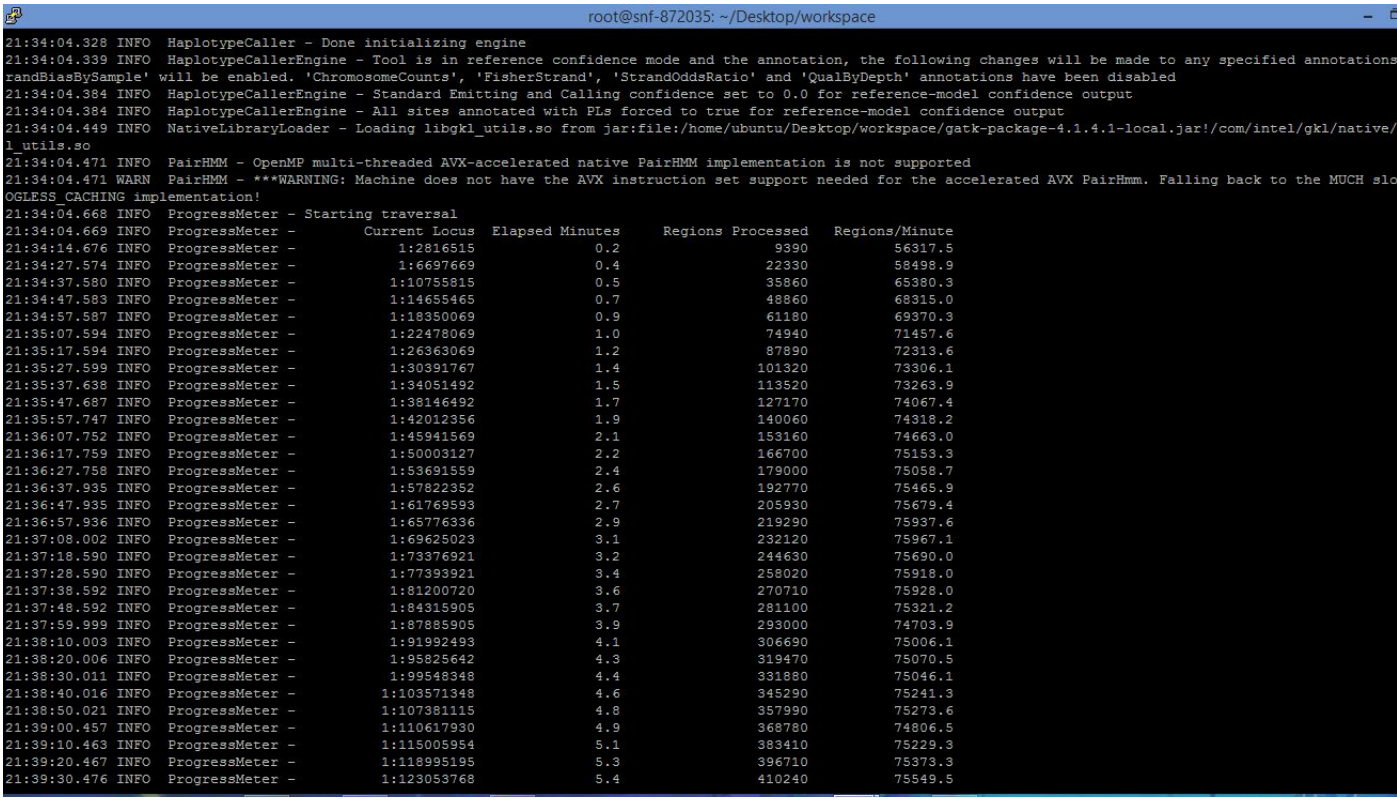
This procedure resulted in the following workspace:



Haplotype Caller

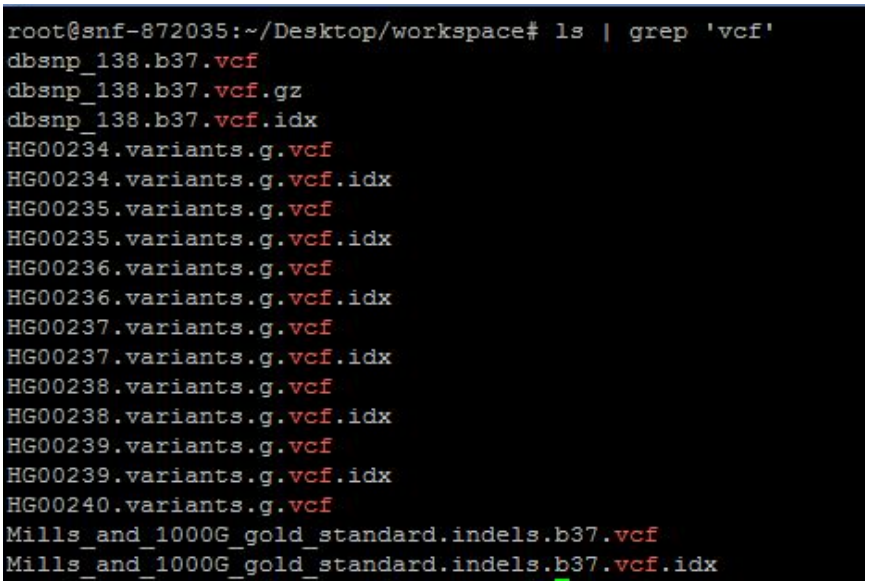
The haplotype caller is the main protagonist of the gatk pipeline. It utilizes pair-Hidden Markov Models on constructed Brujin-graphs in order to identify the haplotypes within the recalibrated read samples.

During the execution, the haplotype caller produces the following output:

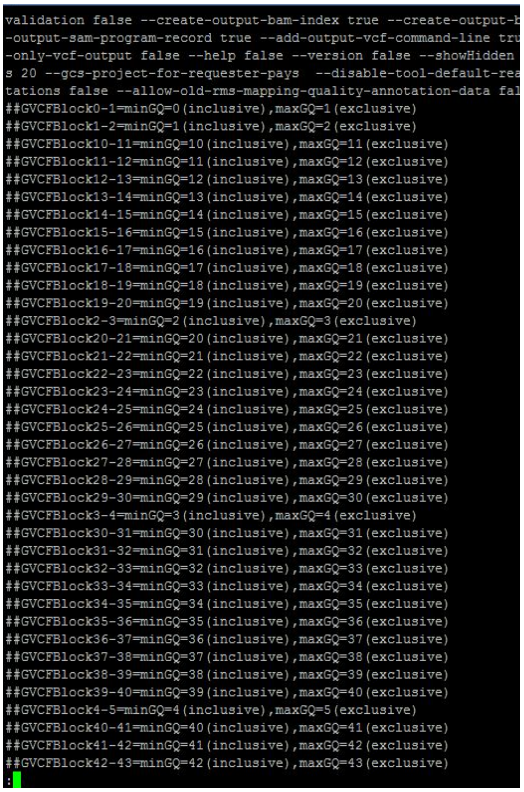


A thing to note is that after reading the warning displayed above, I realized that there exists a compatibility with OpenMP which can take advantage of multiple CPUs through parallel programming to speed up the process. This would be much welcomed since this step takes a long time to complete; for only six samples it has been running for over 24 hours in the okeanos virtual machine.

The Haplotype Caller produces the desired Variant Call Format files (.vcf)

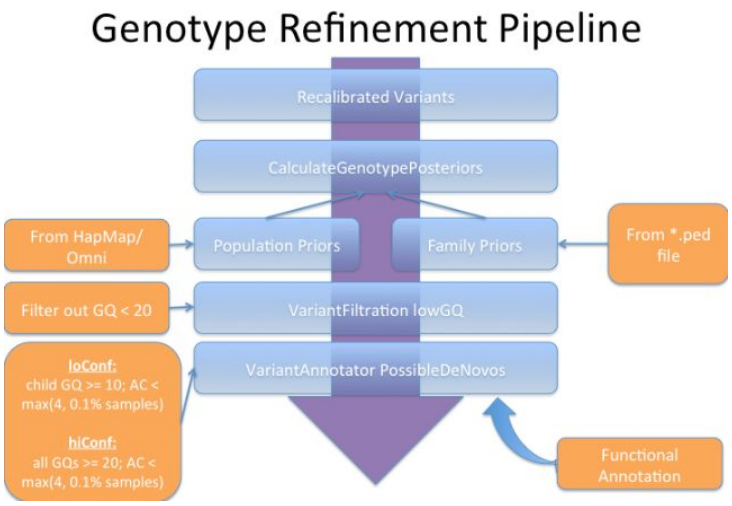


a .vcf file looks like this



Refinement and Final notes

The next step according to the GATK best practices is to filter, refine and annotate the previously produced variants.
https://gatkforums.broadinstitute.org/gatk/discussion/4723/genotype-refinement-workflow
The following is a diagram depicting how this can be achieved in high-level.



Furthermore, if later on, I want to incorporate more samples in my results, I can produce intermediate .gvcf files for only the new samples without having to run the whole procedure from scratch. Then I would consolidate the .gvcfs using gatk GenotypeGVCFs.

In my case, if I were to proceed with the analysis, the following step would be to merge the produced .vcf for each sample and produce a single file containing all identified variants. Then I would try to refine the variants using a pipeline similar to that depicted in the previous picture.

Thank you.
(You can find all commands I run in the log file)