Variant calling assignment report

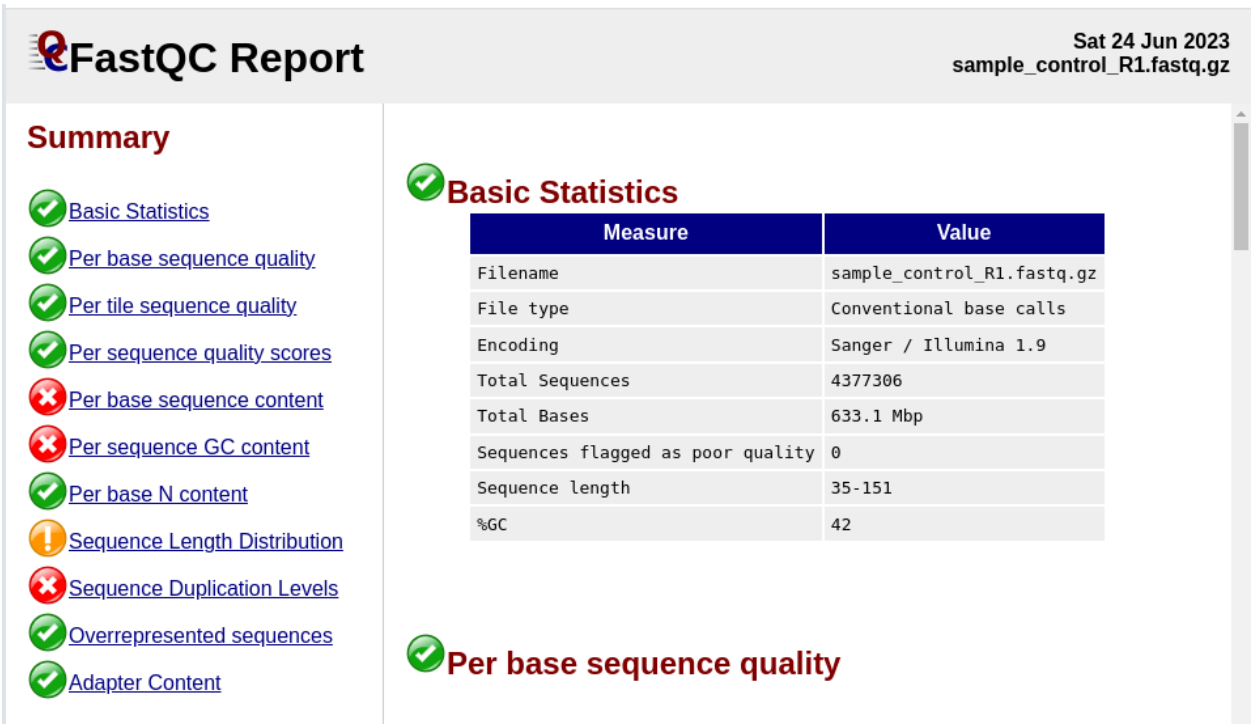
Harald Detering (harald.detering@gmail.com)

28.06.2023

Contents

Quality Control

FastQC reports look healthy. Sequencing adapters were already removed.



FastQC Report

Sat 24 Jun 2023
sample_control_R1.fastq.gz

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	sample_control_R1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	4377306
Total Bases	633.1 Mbp
Sequences flagged as poor quality	0
Sequence length	35-151
%GC	42

Per base sequence quality

Although there are some red flags raised,there is a benign explanation for each of them:

Per base sequence content

The sequence bias in the first 12 positions of the reads is consistent with the bias introduced by Tn5 transposase tagmentation (used e.g. in Nextera library prep), as discussed in the literature.

NOTE: If library prep did not involve Tn5 transposase tagmentation, this could point to a problem with this library.

Per Sequence GC content

The per-sequence GC content distribution shows two peaks, one at ~37% and another one at ~48%. Since the sequencing reads originate from a targeted sequencing experiment, the two peaks most likely correspond to exonic (high GC content) and intronic (low GC content) regions (for reference, see e.g. Amit et al. 2012).

Sequence duplication levels

Duplicated sequences are to be expected, especially in a targeted sequencing experiment, and, more generally, if library prep involved a PCR amplification step. PCR duplicates will be removed in deduplication step during the preprocessing of the analysis pipeline.

NOTE: If library prep did not involve PCR amplification, this could point to a problem with this library.

1. Variant calling analysis

Implementation

I chose to implement the pipeline as a Snakemake workflow for reproducibility, using Conda environments to manage tool dependencies.

The code can be found on GitHub.

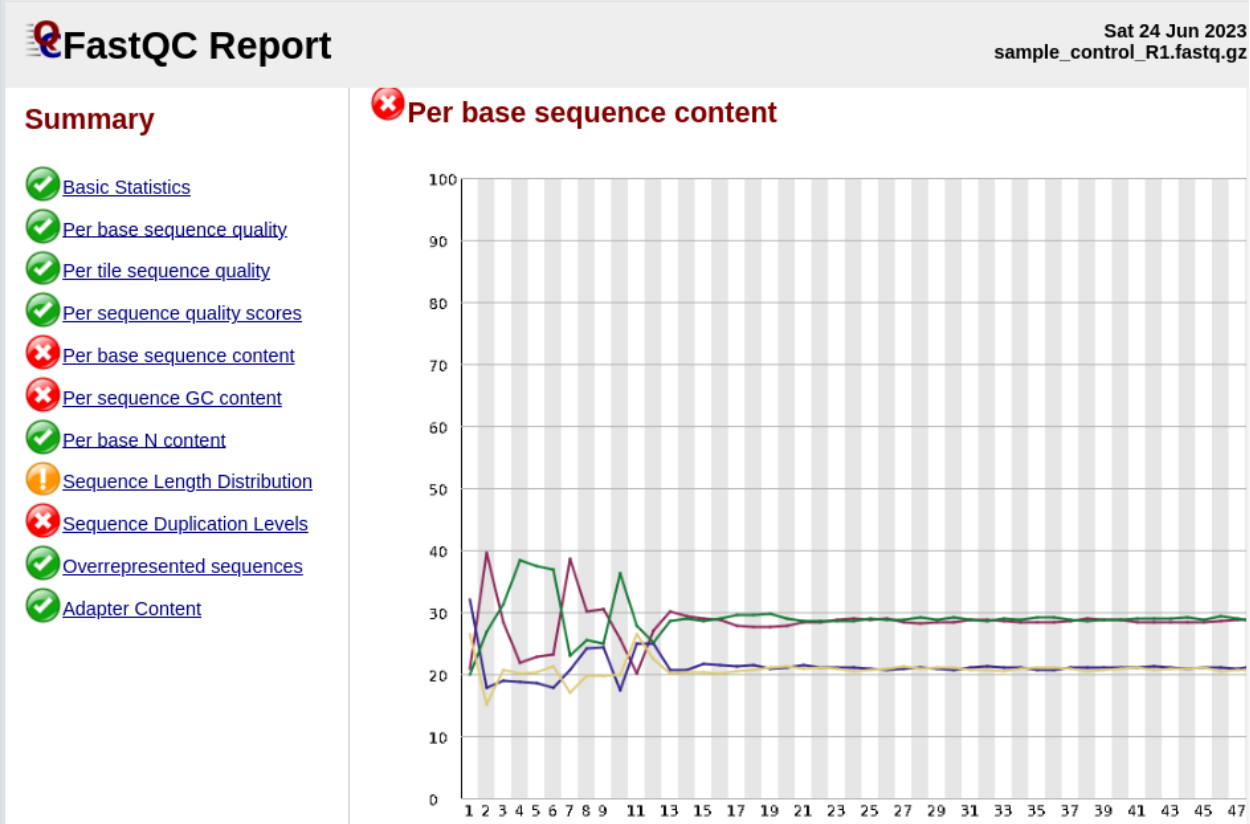


Figure 1: FastQC Per Base Sequence Content

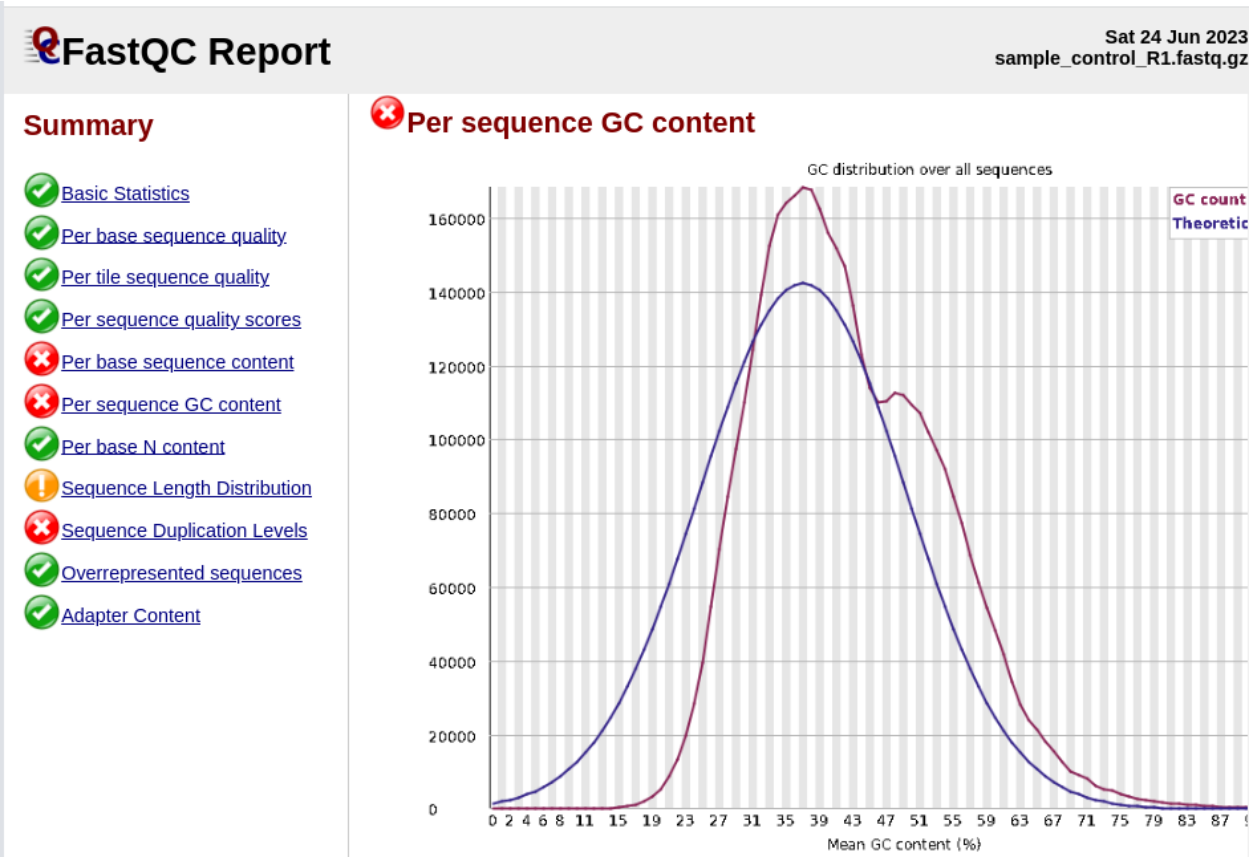


Figure 2: FastQC Per Sequence GC Content

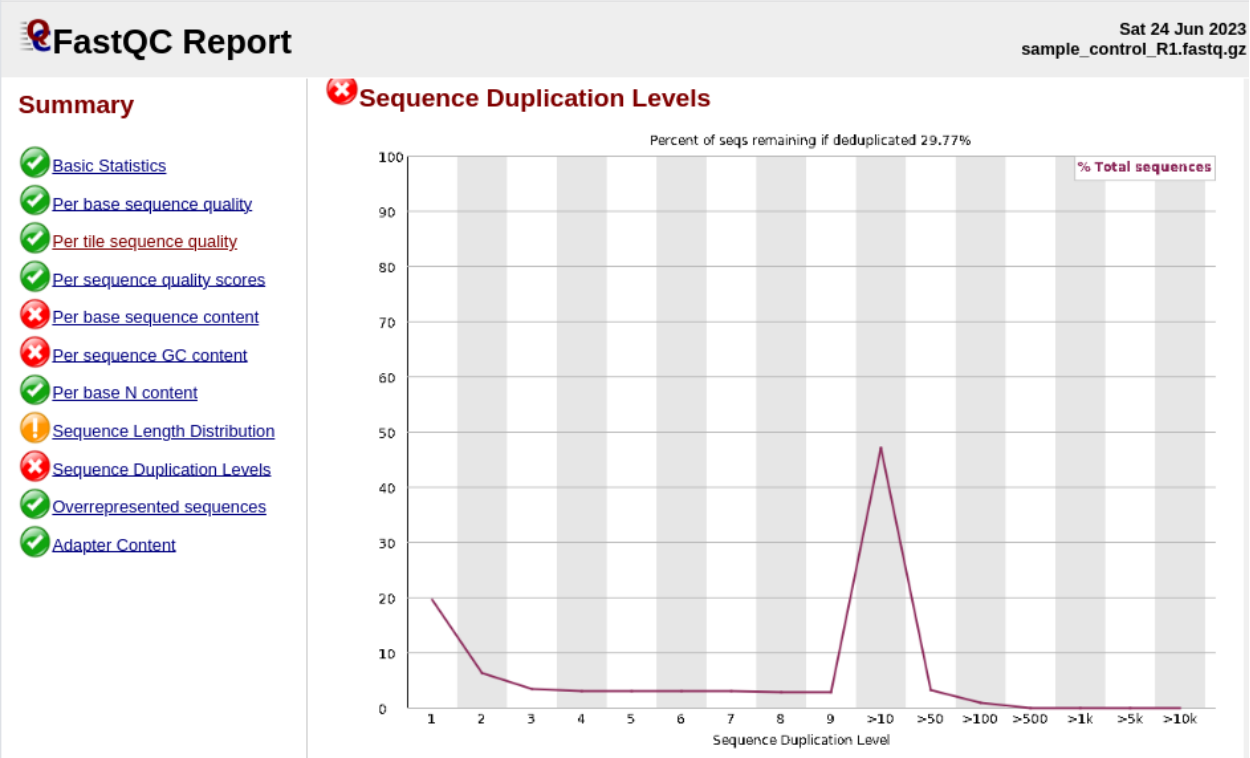


Figure 3: FastQC Sequence Duplication Levels

Results

The variants identified by the variant calling pipeline (for details, see section 3.) are contained in the output file `results/sample_control.recal.vcf`.

2. Analytical performance

The comparison with the true variants (`data/ground_truth.vcf`) yields the following confusion matrix:

```
df_eval <- read.csv('../results/sample_control.eval.csv', header = FALSE, col.names = c('metric', 'valu
df_eval
```

```
##   metric value
## 1     TP    54
## 2     FP    17
## 3     TN 87766
## 4     FN     2
```

Accordingly, we obtain the following performance measures:

```
tp <- df_eval %>% dplyr::filter(metric == 'TP') %>% select(value) %>% pull()
fp <- df_eval %>% dplyr::filter(metric == 'FP') %>% select(value) %>% pull()
tn <- df_eval %>% dplyr::filter(metric == 'TN') %>% select(value) %>% pull()
fn <- df_eval %>% dplyr::filter(metric == 'FN') %>% select(value) %>% pull()

sprintf('Sensitivity / Recall: %.2f', 100 * tp / (tp + fn))
```

```
## [1] "Sensitivity / Recall: 96.43"
```

```
sprintf('Precision: %.2f', 100 * tp / (tp + fp))
```

```
## [1] "Precision: 76.06"
```

```
sprintf('Specificity: %.2f', 100 * tn / (tn + fp))
```

```
## [1] "Specificity: 99.98"
```

Next, let us inspect the FN and FP calls more in detail.

False Negatives (FN)

The following calls were contained in the ground truth variants, but not detected by the variant caller:

```
1   46736848    .   A   T   .   .   .   .   .
13  32930662    .   T   A   .   .   .   .   .
```

1:46736848

For the first FN position (1:46736848), there are simply no ALT bases aligned to the reference. Additionally, the locus is at near the limit of the targeted locus, so there is probably not enough read coverage to capture this particular variant.

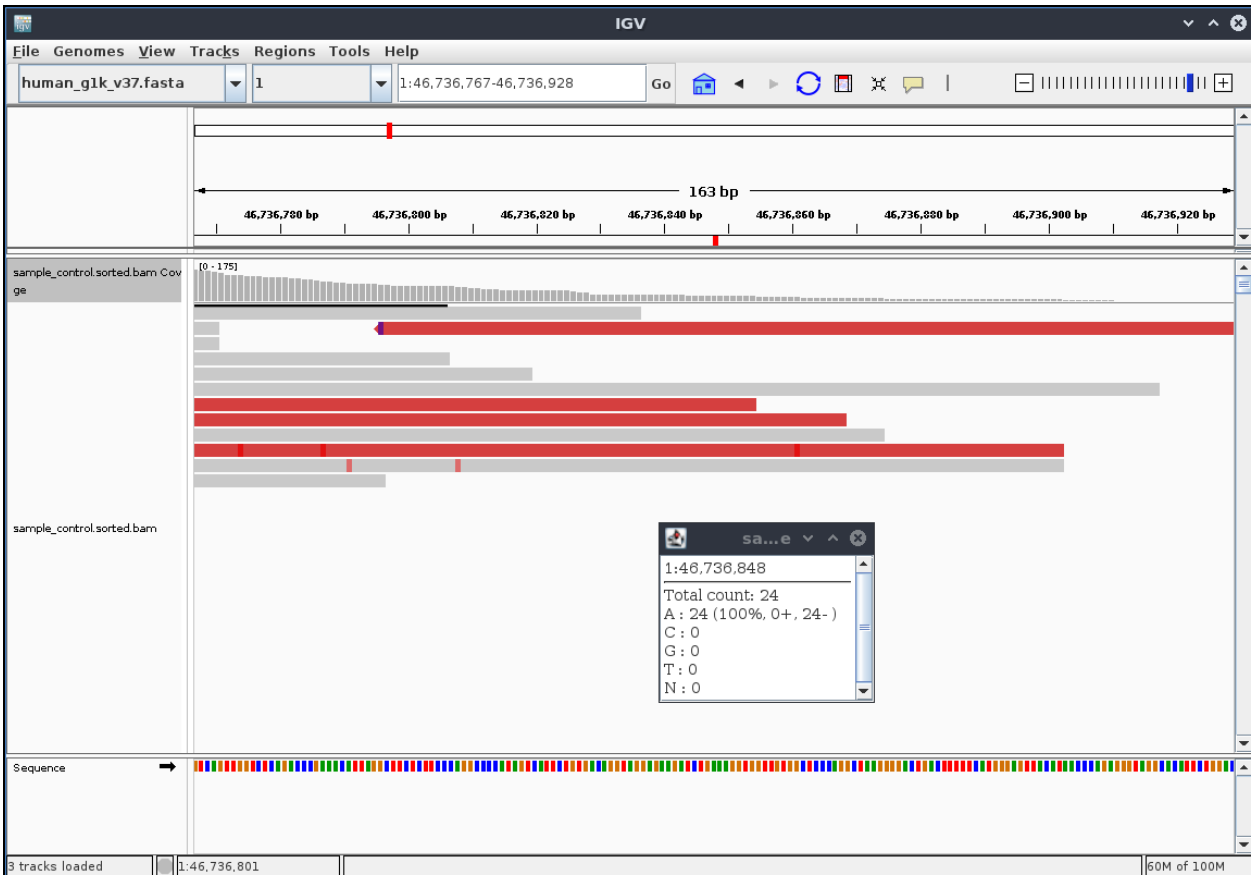


Figure 4: IGV view of FN pos 1:46736848

13:46736848

For the second FN position (13:32930662), read coverage is not an issue. However, the number of reads (3/6226) representing the ALT allele "A" is very low. Due to this extreme allelic imbalance, the variant caller did not detect the variant:

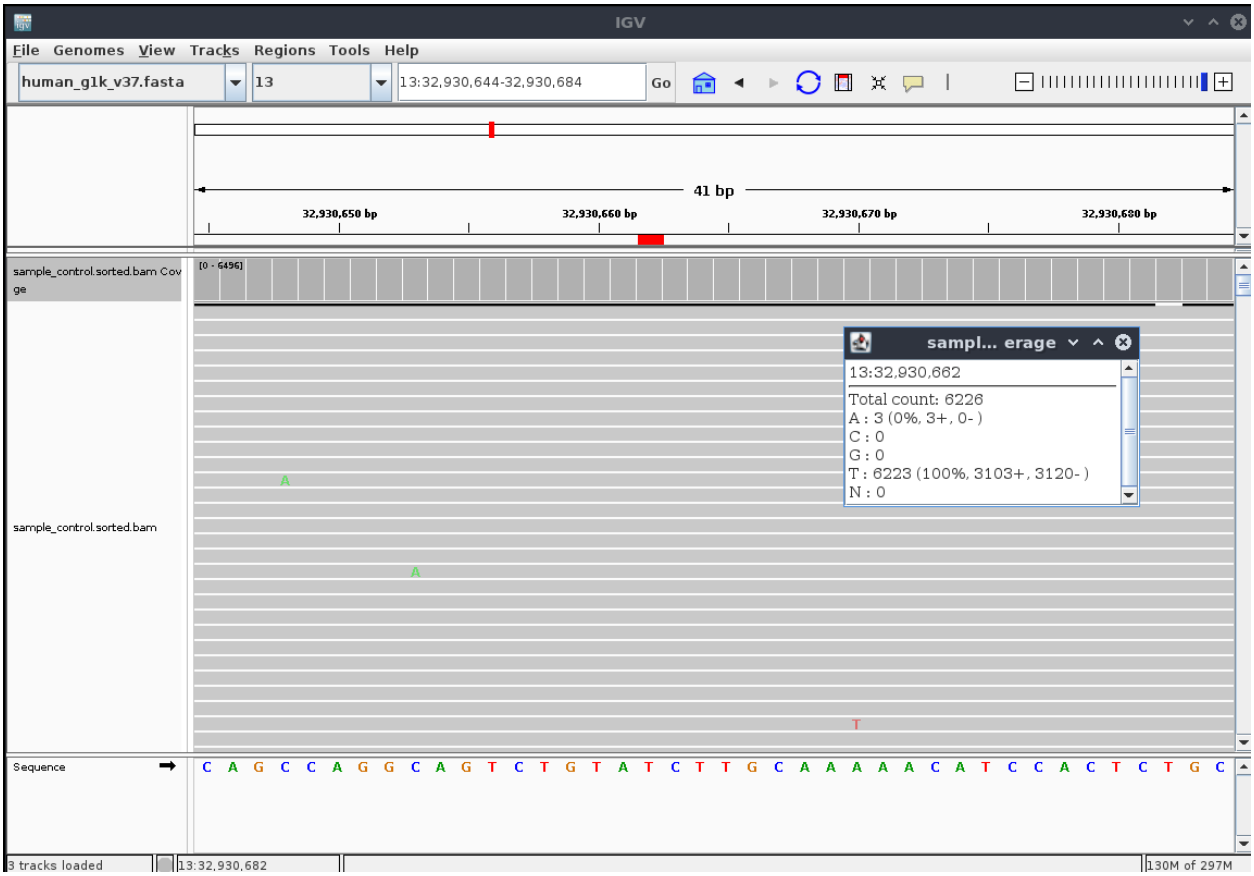


Figure 5: IGV view of FN pos 13:46736848

False Positives (FP)

There were a total of 17 FP variants reported by HaplotypeCaller that were not in the ground truth variants:

```
2   58390218    .   TA  T   934.60  .   AC=1;AF=0.500;AN=2;BaseQRankSum=0.477;DP=3099;ExcessHet=0.0000;
3   10088266    .   G   T   10776.64    .   AC=1;AF=0.500;AN=2;BaseQRankSum=6.316;DP=3663;ExcessHet=0.0
3   10088299    .   C   T   37864.64    .   AC=1;AF=0.500;AN=2;BaseQRankSum=-6.921;DP=3967;ExcessHet=0.
3   10088308    .   T   C   40728.64    .   AC=1;AF=0.500;AN=2;BaseQRankSum=-7.445;DP=4149;ExcessHet=0.
3   10088343    .   A   G   30639.64    .   AC=1;AF=0.500;AN=2;BaseQRankSum=-0.130;DP=4407;ExcessHet=0.
3   10088404    .   C   T   42289.64    .   AC=1;AF=0.500;AN=2;BaseQRankSum=-0.998;DP=3436;ExcessHet=0.
3   10088407    .   AG  A   41939.60    .   AC=1;AF=0.500;AN=2;BaseQRankSum=-5.160;DP=3318;ExcessHet=0.
3   10088409    .   TAAG    T   41162.60    .   AC=1;AF=0.500;AN=2;BaseQRankSum=8.263;DP=3274;ExcessHet=
3   10089723    .   G   A   25892.64    .   AC=1;AF=0.500;AN=2;BaseQRankSum=5.490;DP=3797;ExcessHet=0.0
3   10089738    .   A   G   22897.64    .   AC=1;AF=0.500;AN=2;BaseQRankSum=11.272;DP=3558;ExcessHet=0.
11  108183167   .   A   G   80115.06    .   AC=2;AF=1.00;AN=2;DP=2775;ExcessHet=0.0000;FS=0.000;MLEAC=2
13  32907535    .   CT  C   5877.60  .   AC=1;AF=0.500;AN=2;BaseQRankSum=0.341;DP=3522;ExcessHet=0.0000;
14  68944343    .   CT  C   5903.60  .   AC=1;AF=0.500;AN=2;BaseQRankSum=2.438;DP=1916;ExcessHet=0.0000;
14  69149648    .   G   T   1664.04  .   AC=1;AF=0.500;AN=2;BaseQRankSum=-0.138;DP=563;ExcessHet=0.0000;
14  69149650    .   A   T   2031.64  .   AC=1;AF=0.500;AN=2;BaseQRankSum=-3.169;DP=547;ExcessHet=0.0000;
```

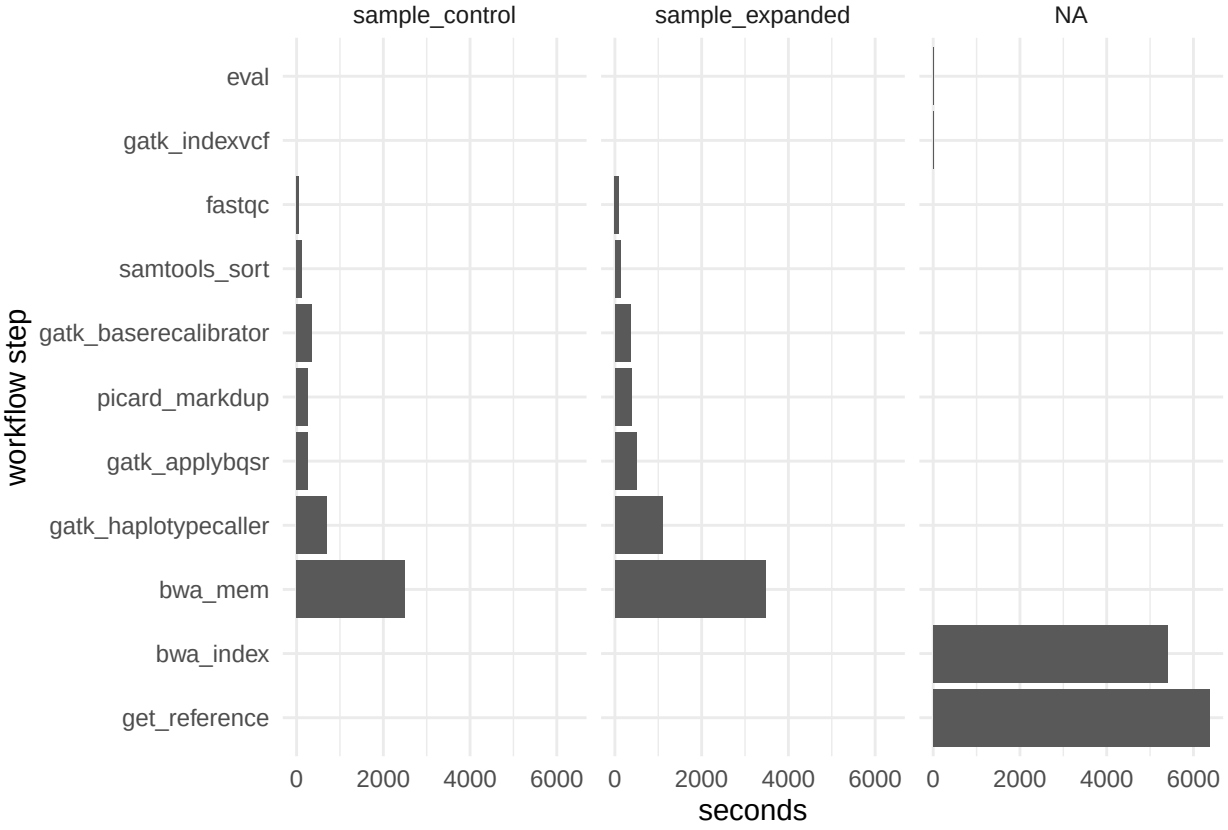
```
14  69149652    .   A   T   362.64  .   AC=1;AF=0.500;AN=2;BaseQRankSum=-6.255;DP=579;ExcessHet=0.0000;
17  59857599    .   C   CA  1021.60 .   AC=1;AF=0.500;AN=2;BaseQRankSum=0.833;DP=2300;ExcessHet=0.0000;
```

All of these positions seem to have decent depth (DP >= ~500). They are probably not in the ground truth set because their AF is too far from 0.5. Still, these variants look like interesting candidates to investigate further, e.g. in the context of subclonal variation.

3. Technical performance

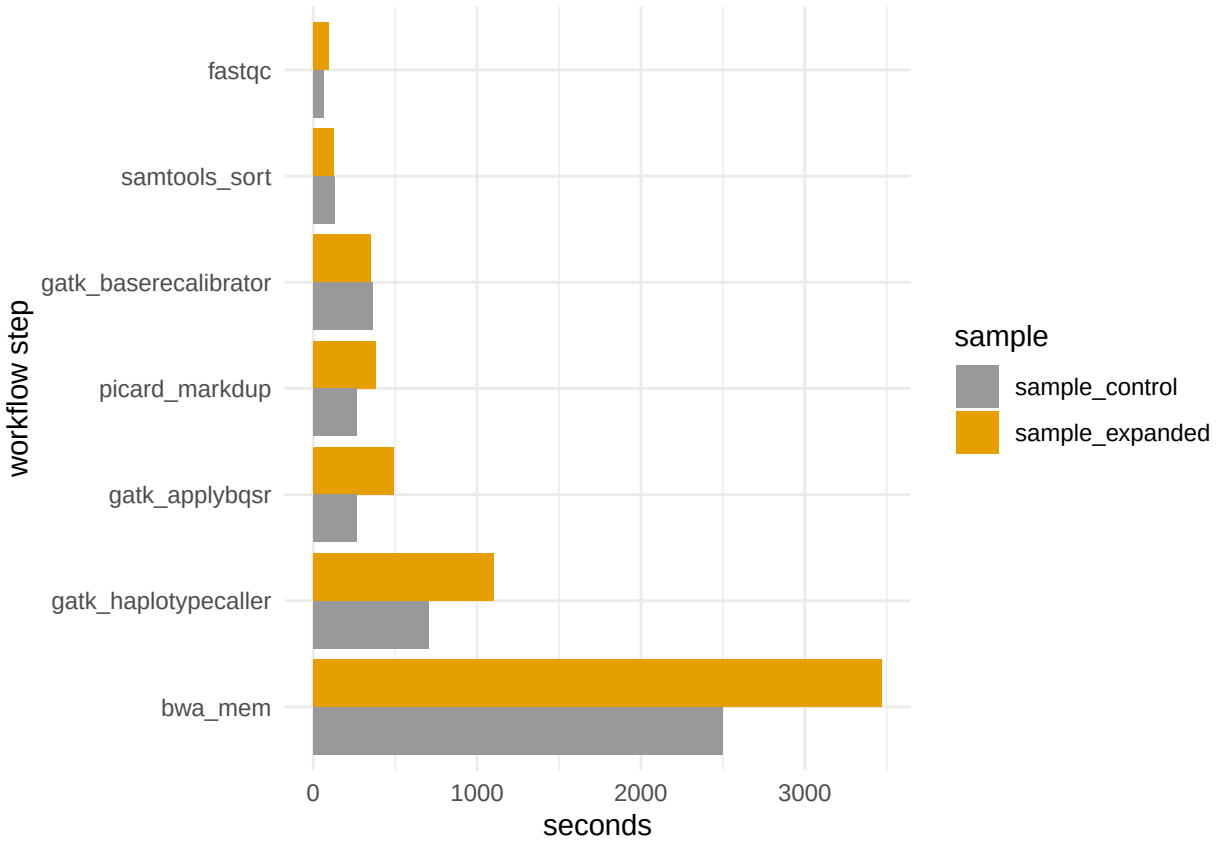
Original dataset

The following plot shows the running time (in seconds) for each workflow step. Where possible, the times are reported by sample (original vs. expanded sample), the common workflow steps are displayed under sample "NA".



It is apparent, that the most time was used for downloading and indexing the reference data (mainly the reference genome). However, this steps will only be carried out once when setting up the variant calling pipeline.

From the performance metrics it is clear that the read mapping (using `bwa mem`) took the majority of the sample-related run time, followed by the variant calling itself (using `gatk HaplotypeCaller`).



The direct comparison between the original and the expanded sample, which has double the number of reads, shows that the major workflow steps (`bwa mem` and `gatk HaplotypeCaller`) scale sub-linearly. Although the comparison has some limitations (1. benchmarking runs were not performed on a dedicated machine and 2. the expanded dataset is quite synthetic), this is an encouraging result that suggests that the variant calling pipeline will scale well with larger datasets.

Expanded dataset

4. Pipeline documentation

A graphical summary of the most important pipeline steps is shown in the following flow chart:

Reference datasets

In the assignment description it says that the human reference "hg19" was used. However, the VCF file `ground_truth.vcf` mentions `human_g1k_v37.fasta`, which is consistent with the genomic coordinates used ("b37" uses chromosome names: "1", "2", etc., whereas "hg19" uses "chr1", "chr2", etc.). Therefore I downloaded the reference files from the "b37" dataset (gs://gatk-legacy-bundles/b37). For the BQSR I also downloaded the reference VCF files for known SNPs and Indels.

Filename	Description
human_g1k_v37.fasta	Human reference genome sequence
dbsnp_138.b37.vcf	Reference SNPs from dbSnp v. 138
1000G_phase3_v4_20130502.sites.vcf	Reference SNPs from 1000Genomes
Mills_and_1000G_gold_standard.indels.b37.vcf	Reference Indels from Mills and 1000Genomes

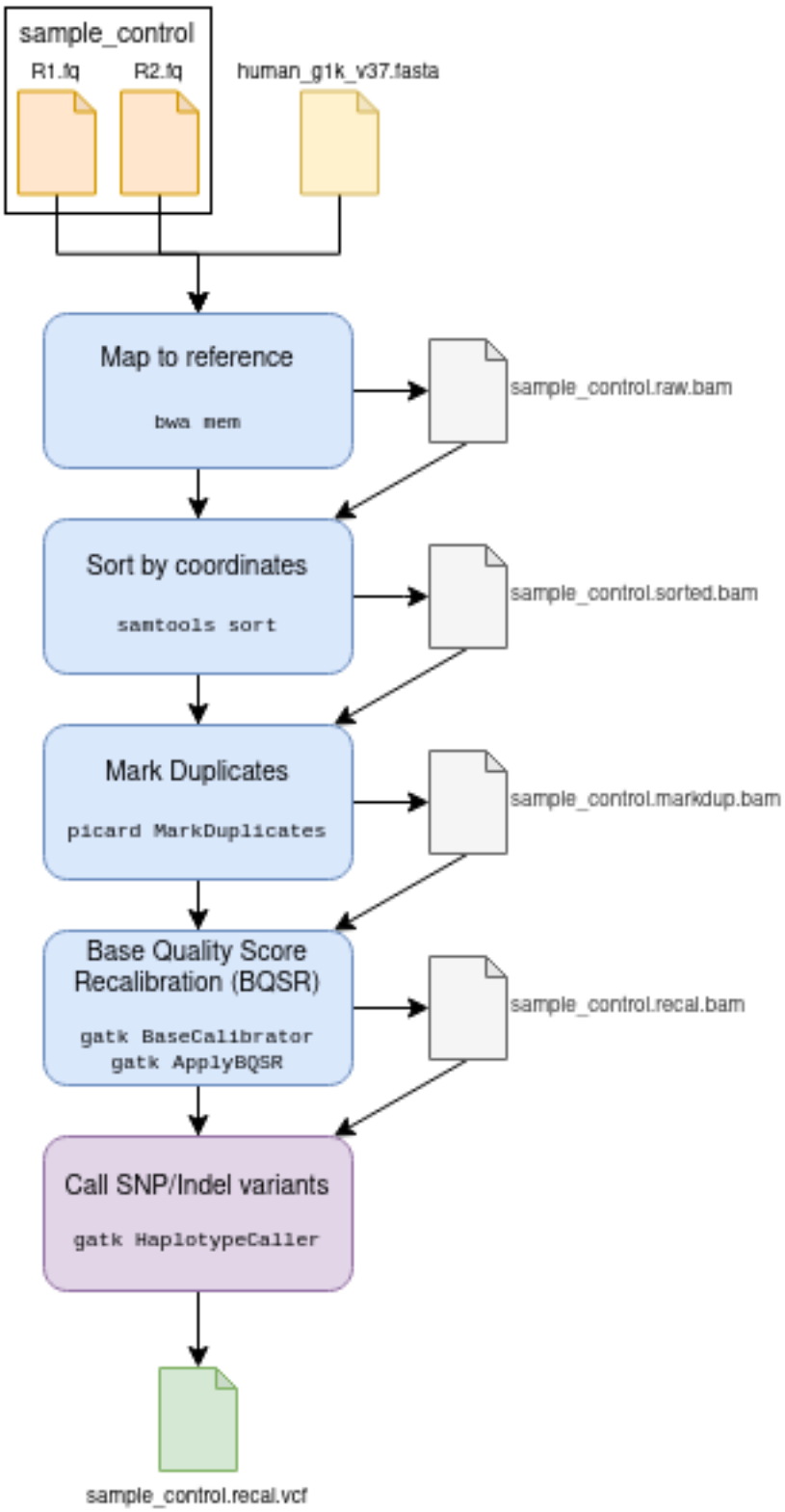


Figure 6: Diagram of Variant Calling Pipeline

Indexing

The reference FASTA and VCFs were indexed using Picard CreateSequenceDictionary and GATK IndexFeatureFile, respectively. A BWT index for the reference FASTA was created using bwa index as a prerequisite for read mapping with bwa mem.

Read mapping

Raw paired-end reads were mapped for one sample (`sample_control`) against the reference using bwa mem, with standard parameters and adding a ReadGroup.

PARAMETERS:

- bwa mem:
  - -R '@RG\tID:1\tSM:sample_control\tPL:illumina\tPU:Lane1\tLB:Targeted'

INPUT:

- reference FASTA: `human_g1k_v37.fasta`
- reference FASTA index files: `.amb`, `.ann`, `.bwt`, `.fai`, `.pac`, `.sa`
- sequencing reads:
  - `sample_control_R1.fastq.gz`
  - `sample_control_R2.fastq.gz`

OUTPUT:

- raw read mappings: `sample_control.raw.sam`

Read mapping sorting

Read mappings were sorted by genomic coordinates using samtools sort and indexed using samtools index.

PARAMETERS:

(default params)

INPUT:

- raw read mapping: `sample_control.raw.sam`

OUTPUT:

- sorted read mappings: `sample_control.sorted.bam`
- read mapping index: `sample_control.sorted.bam.bai`

Mark Duplicates

Since we are working with targeted sequencing, it is important to mark PCR duplicates, so that they can be accounted for during variant calling. Therefore, duplicate alignments were marked using Picard MarkDuplicates.

PARAMATERS:

(default params)

INPUT:

- sorted read mappings: `sample_control.sorted.bam`

OUTPUT:

- read mappings with duplicates marked: `sample_control.markdup.bam`
- duplicates metrics: `sample_control.markdup.metrics.txt`

Base Quality Score Recalibration (BQSR)

BQSR can help reduce systematic base calling biases due to the sequencing technology used. BQSR is performed in two steps: 1) calculate recalibration table and 2) recalibrate base quality scores.

Calculate Recalibration Table

To calculate the recalibration table in the context of known variant loci, GATK BaseRecalibrator is used.

PARAMETERS:

(default params)

INPUT:

- read mappings: `sample_control.markdup.bam`
- reference FASTA: `human_g1k_v37.fasta`
- reference SNPs from dbSnp: `dbsnp_138.b37.vcf`
- reference SNPs from 1000Genomes: `1000G_phase3_v4_20130502.sites.vcf`
- reference Indels: `Mills_and_1000G_gold_standard.indels.b37.vcf`

OUTPUT:

- recalibration table: `sample_control.recal.table`

Recalibrate Base Quality Scores

To perform the BQSR, GATK ApplyBQSR is used.

PARAMETERS:

(default params)

INPUT:

- reference FASTA: `human_g1k_v37.fasta`
- read mappings: `sample_control.markdup.bam`
- recalibration table: `sample_control.recal.table`

OUTPUT:

- recalibrated read mappings: `sample_control.recal.bam`

Variant detection

The detection of SNPs and Indels from the preprocessed read alignments was done using GATK HaplotypeCaller.

PARAMETERS:

(default params)

INPUT:

- read mappings: `sample_control.recal.bam`
- reference FASTA: `human_g1k_v37.fasta`
- genomic regions: `target_regions.bed`

OUTPUT:

- detected variants: `sample_control.recal.vcf`

Observation

A total of 71 variants were detected, 18 of which were indels. These are slightly more variants than in the ground truth VCF (56 total, 12 indels).