Introduction to Bioinformatics

BLG 348E

Term Project Report

Rengin Helin Yalçın
yalcinr22@itu.edu.tr

Faculty of Computer and Informatics Engineering
Department of Computer Engineering
Date of submission: 26.12.2024

1. Introduction

1.1. Abstract

This report aims to evaluate and compare the performance of multiple variant calling pipelines which are Mutect, Strelka and SomaticSniper, and alignment algorithms, BWA and Bowtie.

1.2. Introduction

DNA sequencing is the process of determining the precise order of nucleotides in a DNA molecule. This technology allows researchers to decode the genetic information contained within an organism's genome, which dictates biological structure, function, and inheritance. Modern sequencing technologies have paved the way for comprehensive genomic studies, such as variant detection, which involves identifying genetic differences like single nucleotide polymorphisms (SNPs) and small insertions or deletions (indels).

In the context of variant calling, the reliability of results depends heavily on the alignment algorithms and variant calling tools used. Alignment tools such as BWA and Bowtie are widely employed to map sequencing reads to reference genomes. BWA excels at aligning short reads with high accuracy and is frequently used in whole-genome and exome sequencing pipelines. Bowtie, known for its speed and low memory usage, is optimized for aligning large datasets and is commonly employed in RNA sequencing and other high-throughput sequencing applications.

Once aligned, variant callers like Strelka, SomaticSniper, and Mutect are utilized to identify genetic variations. Strelka is a highly sensitive tool for detecting somatic and germline variants, particularly in cancer studies. It is designed to work efficiently with paired tumor-normal samples, making it a popular choice for cancer genomics research. SomaticSniper specializes in detecting somatic mutations by comparing tumor and normal samples, offering reliable results in identifying cancer-specific genetic alterations. Mutect, a widely used variant caller, excels at identifying somatic mutations in tumor samples, particularly in low-frequency variants.

1.2.1. Computer Properties

The computer I used during this project, ASUS VivoBook X571GT, had the following properties:

- Windows 11 OS
- Intel(R) Core(TM) i7-9750H CPU

- 16,0 GB RAM
- 457 GB Total Storage

I used Windows Subsystem for Linux to run the pipelines, which ran Ubuntu 24.04.1 LTS.

As the first step during this project, after downloading the necessary files I ensured the integrity of the data.

```
(base) rengin@RenginHelin:~/data$ sha1sum -c term_project_data.sha1
1000G_phase1.snps.high_confidence.hg38.vcf.gz: OK
1000G_phase1.snps.high_confidence.hg38.vcf.gz.tbi: OK
Homo_sapiens_assembly38.1.bt2: OK
Homo_sapiens_assembly38.2.bt2: OK
Homo_sapiens_assembly38.3.bt2: OK
Homo_sapiens_assembly38.4.bt2: OK
Homo_sapiens_assembly38.dbsnp138.elsites: OK
Homo_sapiens_assembly38.dbsnp138.vcf: OK
Homo_sapiens_assembly38.dbsnp138.vcf.idx: OK
Homo_sapiens_assembly38.dict: OK
Homo_sapiens_assembly38.elfasta: OK
Homo_sapiens_assembly38.fasta: OK
Homo_sapiens_assembly38.fasta.64.alt: OK
Homo_sapiens_assembly38.fasta.64.amb: OK
Homo_sapiens_assembly38.fasta.64.ann: OK
Homo_sapiens_assembly38.fasta.64.bwt: OK
Homo_sapiens_assembly38.fasta.64.pac: OK
Homo_sapiens_assembly38.fasta.64.sa: OK
Homo_sapiens_assembly38.fasta.bwt.2bit.64: OK
Homo_sapiens_assembly38.fasta.fai: OK
Homo_sapiens_assembly38.rev.1.bt2: OK
Homo_sapiens_assembly38.rev.2.bt2: OK
Mills_and_1000G_gold_standard.indels.hg38.vcf.gz: OK
Mills_and_1000G_gold_standard.indels.hg38.vcf.gz.tbi: OK
SRR7890850_1.fastq.gz: OK
SRR7890850_2.fastq.gz: OK
SRR7890851_1.fastq.gz: OK
SRR7890851_2.fastq.gz: OK
(base) rengin@RenginHelin:~/data$
```

Figure 1.1: Ensuring Data Integrity

After completing this step, I moved on to pipelines.

2. Results

At first, I ran the trimming pipelines and successfully generated trimmed fastq files. However, the alignment process created many errors so I moved on to using the BAM files that were shared with us.

Mutect took around 3 hours to execute, whereas other tools took less than an hour. To run Strelka, I used docker and executed the pipelines with the following command:

```
>>$ sudo docker run -it \
    -v /home/rengin/cosap_data/:/cosap_data \
    -v /home/rengin/cosap_data/:/workdir \
    -v /var/run/docker.sock:/var/run/docker.sock \
    itubioinformatics/cosap python pipeline.py
```

After running the pipelines, the following steps were taken for filtering:

- gatk UpdateVCFSequenceDictionary on SomaticSniper VCFs to fix the headers of the file.
- bcftools +fixploidy to fix incorrect ploidy information.
- BED Filtering to direct the analysis to a genomic region.
- bcftools to filter VCF files based on the “FILTER” column.
- Final filtering with snpfilter.pl for SomaticSniper VCFs.

```
(cosap) rengin@RenginHelin:~/last$ bcftools view -f PASS "strelka_bowtie_bedfil.vcf.recode.vcf" -o "filtered_strelka_bowtie.vcf"
(cosap) rengin@RenginHelin:~/last$ bcftools view -f PASS "strelka_bwa_bedfil.vcf.recode.vcf" -o "filtered_strelka_bwa.vcf"
(cosap) rengin@RenginHelin:~/last$ ls
filtered_strelka_bowtie.vcf   sniper_bowtie_bedfil.vcf.recode.vcf  strelka_bowtie.vcf                  strelka_bwa_bedfil.vcf.log
filtered_strelka_bwa.vcf      sniper_bwa.vcf                      strelka_bowtie_bedfil.vcf.log       strelka_bwa_bedfil.vcf.recode.vcf
sniper_bowtie.vcf             sniper_bwa_bedfil.vcf.log           strelka_bowtie_bedfil.vcf.recode.vcf
sniper_bowtie_bedfil.vcf.log  sniper_bwa_bedfil.vcf.recode.vcf    strelka_bwa.vcf
(cosap) rengin@RenginHelin:~/last$
```

Figure 2.1: bcftools to filter VCF files based on the “FILTER” column.

On Galaxy, marking duplicates step failed for the normal bam of BWA. Therefore I only proceeded with Bowtie and generated 1 VCF file.

After generating (1 sample x 2 recalibration conditions x 2 mapper x 3 variant caller) 12 pipelines using COSAP and 1 pipeline using Galaxy, I moved on to analysis.

Here are the precision, recall, accuracy and F1 score values based on the results:

Table 2.1: Precision, Recall, Accuracy and F1-Score for Variant Calling Pipelines

Pipeline	TP	FP	FN	Precision	Recall	F1-Score	Accuracy
Strelka + BWA + Base Recalibration	936	1877	225	0.3327	0.8062	0.4711	0.3081
Strelka + Bowtie + Base Recalibration	826	723	335	0.5332	0.7115	0.6096	0.4384
Strelka + BWA	826	723	335	0.5332	0.7115	0.6096	0.4384
Strelka + Bowtie	826	723	335	0.5332	0.7115	0.6096	0.4384
SomaticSniper + BWA + Base Recalibration	854	7433	307	0.1031	0.7356	0.1808	0.0994
SomaticSniper + Bowtie + Base Recalibration	781	4742	380	0.1414	0.6727	0.2337	0.1323
SomaticSniper + BWA	870	8527	291	0.0926	0.7494	0.1648	0.0898
SomaticSniper + Bowtie	804	6054	357	0.1172	0.6925	0.2005	0.1114
Mutect + BWA + Base Recalibration	909	470	252	0.6592	0.7829	0.7157	0.5573
Mutect + Bowtie + Base Recalibration	876	358	285	0.7099	0.7545	0.7315	0.5767
Mutect + BWA	907	528	254	0.6321	0.7812	0.6988	0.5370
Mutect + Bowtie	858	327	303	0.7241	0.7390	0.7315	0.5766
Galaxy + Bowtie	832	415	329	0.6672	0.7166	0.6910	0.5279

Precision measures how many of the detected variants are correct. High precision means the tool avoids false positives, but it does not consider missed variants (false negatives). Recall measures how many true variants were correctly identified. High recall indicates that the tool detects most variants, but it does not penalize false positives. The F1 Score is the harmonic mean of Precision and Recall. It balances both metrics and is useful when false positives and false negatives are equally important. Accuracy measures the overall correctness of predictions.

These results indicate that Mutect is effective for general variant calling with high F1 Scores. SomaticSniper, on the other hand, seems unreliable in this dataset. The choice of alignment method (e.g., bwa vs. bowtie) and variant calling tool significantly impacts performance. It seems that Bowtie showed a much better performance for all variant calling pipelines. Base recalibration is also necessary for better results according to the numbers, except, Strelka performed poorly with BWA and base recalibration conditions compared to other pipelines of it.

The calculations were done the way it can be seen in the Figure 2.2. True negative values are taken as zero.

```
def calculate_metrics(ground_truth_set, predicted_set):
    TP = len(ground_truth_set & predicted_set)  # True positives: intersection
    FP = len(predicted_set - ground_truth_set)   # False positives: predicted but not in ground truth
    FN = len(ground_truth_set - predicted_set)   # False negatives: in ground truth but not predicted

    precision = TP / (TP + FP) if (TP + FP) > 0 else 0
    recall = TP / (TP + FN) if (TP + FN) > 0 else 0
    f1_score = 2 * (precision * recall) / (precision + recall) if (precision + recall) > 0 else 0
    accuracy = TP / (TP + FP + FN) if (TP + FP + FN) > 0 else 0

    return precision, recall, f1_score, accuracy
```

Figure 2.2: Code Snippet for Metric Calculations

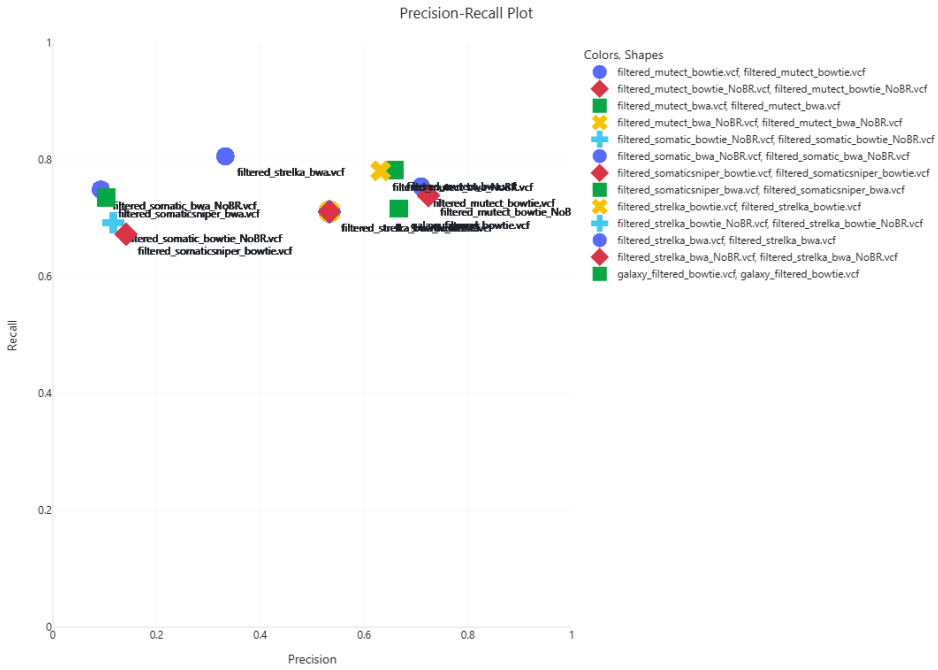

Figure 2.3: Precision-Recall Plot

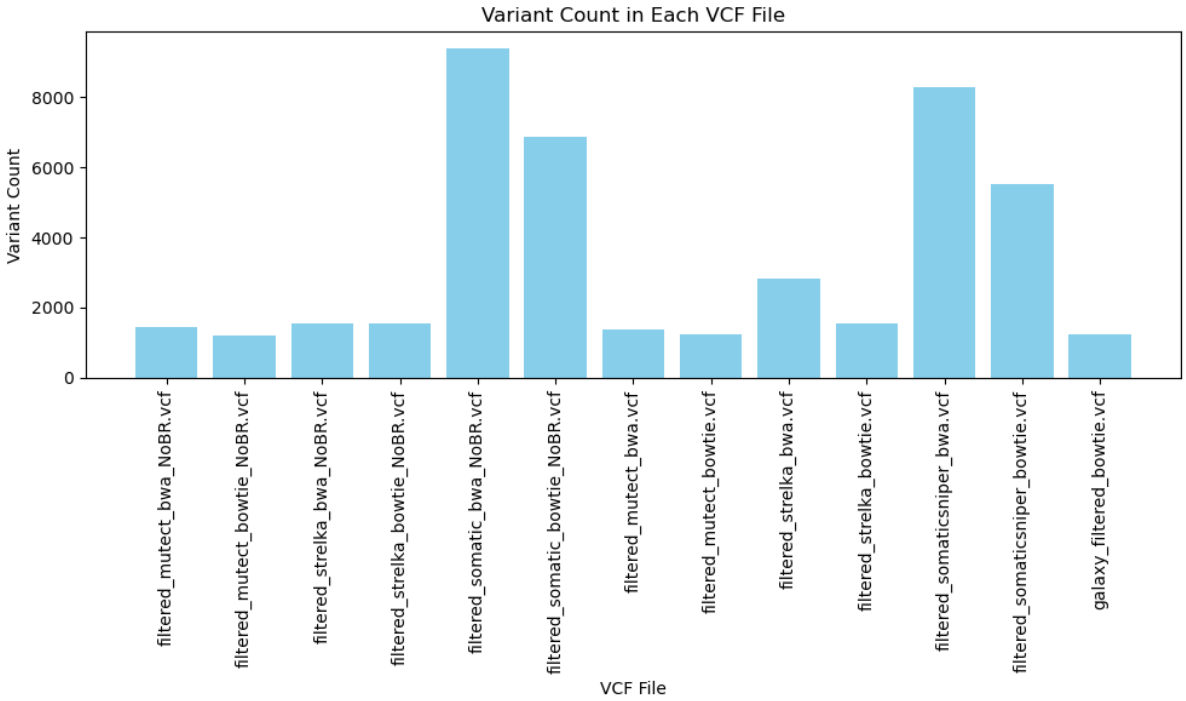

Figure 2.4: Variant Count

The variant count showed that there is significant variation in the number of variants across different files. A high variant count indicates that many genetic variations were identified in the VCF file. The SomaticSniper files generally show higher variant counts compared to the Mutect and Strelka files, regardless of the alignment tool. Using BWA

results in more variants for SomaticSniper. However, the difference between BWA and Bowtie methods seems minimal for Mutect and Strelka.

```
(cosap) rengin@RenginHelin:~/last$ python variant.py
File: filtered_mutect_bwa_NoBR.vcf, Variant Count: 1435
File: filtered_mutect_bowtie_NoBR.vcf, Variant Count: 1185
File: filtered_strelka_bwa_NoBR.vcf, Variant Count: 1549
File: filtered_strelka_bowtie_NoBR.vcf, Variant Count: 1549
File: filtered_somatic_bwa_NoBR.vcf, Variant Count: 9397
File: filtered_somatic_bowtie_NoBR.vcf, Variant Count: 6858
File: filtered_mutect_bwa.vcf, Variant Count: 1379
File: filtered_mutect_bowtie.vcf, Variant Count: 1234
File: filtered_strelka_bwa.vcf, Variant Count: 2813
File: filtered_strelka_bowtie.vcf, Variant Count: 1549
File: filtered_somaticsniper_bwa.vcf, Variant Count: 8287
File: filtered_somaticsniper_bowtie.vcf, Variant Count: 5523
File: galaxy_filtered_bowtie.vcf, Variant Count: 1247
```

Figure 2.5: Variant Count by Number

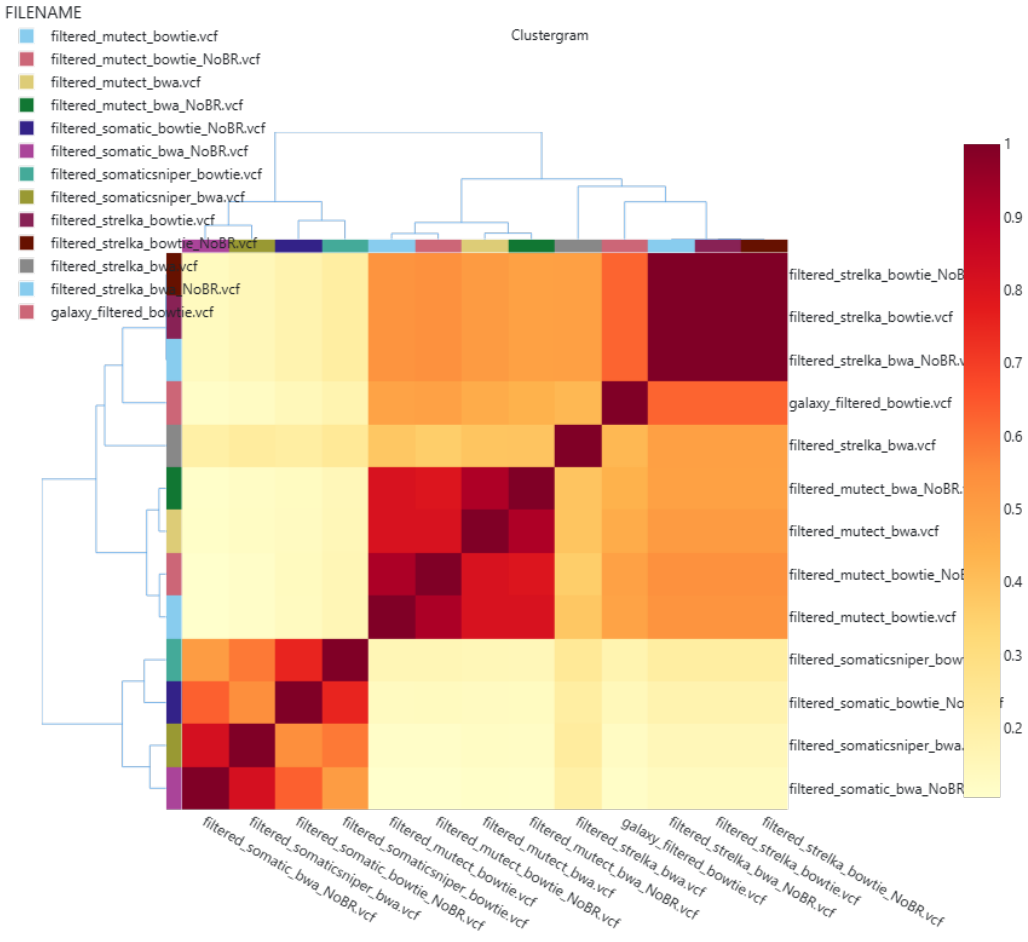

Figure 2.6: Clustergram

2.1. PCA Plot

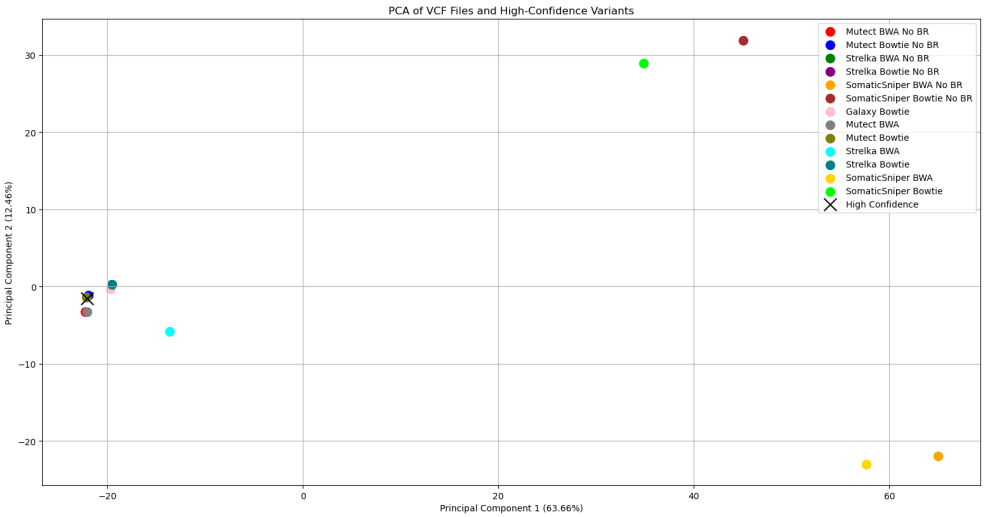

Figure 2.7: PCA Plot

Principal Component 1 (PC1) accounts for 63.66% of the variance in the data, meaning most of the differences among the samples are captured by this axis. The two principal components together explain about 76.12% (63.66% + 12.46%) of the variability in the data. This means the majority of the important patterns in the dataset are represented in this 2D plot.

Tools like Strelka and Mutect for both BWA and Bowtie seem more consistent with the High Confidence reference compared to SomaticSniper. For SomaticSniper, base recalibration significantly changes the results. Points are even further away when the alignment tool is also different.

```
(cosap) rengin@RenginHelin:~/last$ python pca.py
PCA Coordinates:
Mutect BWA No BR: PC1 = -22.295, PC2 = -3.273
Mutect Bowtie No BR: PC1 = -21.931, PC2 = -1.147
Strelka BWA No BR: PC1 = -19.567, PC2 = 0.244
Strelka Bowtie No BR: PC1 = -19.567, PC2 = 0.244
SomaticSniper BWA No BR: PC1 = 64.973, PC2 = -21.905
SomaticSniper Bowtie No BR: PC1 = 45.015, PC2 = 31.910
Galaxy Bowtie: PC1 = -19.682, PC2 = -0.284
Mutect BWA: PC1 = -22.055, PC2 = -3.265
Mutect Bowtie: PC1 = -22.155, PC2 = -1.329
Strelka BWA: PC1 = -13.616, PC2 = -5.832
Strelka Bowtie: PC1 = -19.567, PC2 = 0.244
SomaticSniper BWA: PC1 = 57.654, PC2 = -23.000
SomaticSniper Bowtie: PC1 = 34.853, PC2 = 28.936
High Confidence: PC1 = -22.061, PC2 = -1.542
```

Figure 2.8: PCA Coordinates

3. Discussion

From the analysis, it is clear that the Mutect pipelines perform the best overall. They have a good balance between identifying true variants (high recall) and minimizing incorrect calls (high precision). This makes them reliable for variant detection tasks.

The Galaxy Strelka pipeline is another good choice, offering competitive performance, with fewer false positives compared to others. It could be a simpler alternative for users looking for decent accuracy without too many errors.

On the other hand, SomaticSniper pipelines performed the worst, with very high numbers of false positives. This means they flagged many variants incorrectly, making their results less trustworthy. They would need additional filtering or adjustments to become usable.

The Strelka pipelines sit somewhere in the middle, with moderate performance. While they are decent at finding true variants, their accuracy is lower due to the higher number of false positives compared to Mutect.

To improve these results further, combining the strengths of multiple pipelines or improving settings in weaker pipelines could lead to even better variant detection.