Istanbul Technical University
Faculty of Computer and Informatics
Computer Engineering Department

Introduction to Bioinformatics
Term Project
Report

Aybike Zeynep Taşkın
150190004
Bilal İhsan Tuncer
150190089

December 19th, 2023

Contents

1 Abstract

In this comprehensive analysis, 12 next-generation sequencing (NGS) pipelines, integrating different combinations of mappers (BWA and Bowtie) and variant callers (SomaticSniper, Mutect, and Strelka), were evaluated using the Comparative Sequencing Analysis Platform (CoSAP). The study focused on assessing the performance of these pipelines with and without base recalibration. Performance metrics like precision, recall, F1-score, and accuracy were meticulously calculated for each pipeline, providing a quantitative assessment of their variant calling efficacy. The analysis was enriched with the use of Jaccard distance to construct heatmaps, illustrating the similarities and differences in pipeline outputs. Additionally, histograms offered a visual distribution of the performance metrics across configurations. A critical aspect of the study was employing Principal Component Analysis (PCA) for a nuanced, lower-dimensional representation of the relationships between pipelines. This analytical approach highlighted the influences of algorithmic choices on variant calling, offering valuable insights into optimizing NGS pipeline configurations for genomic analysis.

2 Introduction

DNA sequencing is the process of determining the precise order of nucleotides within a DNA molecule. It is a fundamental technique in molecular biology and genomics, providing crucial information about the genetic code and the specific sequence of genes and other functional elements in an organism's genome. One of the most important technological advancements of the century in the field of biological sciences, is next-generation sequencing, or NGS. For good reason, these are also known as high-throughput sequencing technologies since they produce enormous volumes of data that must be analyzed. The primary cause of computational biology's transformation into a big-data field is NGS. This is a field that, more than anything else, calls for excellent bioinformatics approaches [1].

In our project, we needed to compare performances of various next generation sequencing data processing algorithms, so we used CoSAP as our pipeline creation tool as it was implied by the project pdf. First, we made sure that we had sufficient hardware requirements. We used ASUS ROG laptop with 16GB RAM for our project. It has Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz processor. It has Windows 10 operating system. But we used Windows Subsystem for Linux (WSL) for our implementations. The computer had total of 476GB space but the empty space was barely enough for our project.

We started by installing the necessary packages from the github repository of the MBaysanLab. Then, we downloaded all the necessary data. Initially, we downloaded 2 FASTQ files. FASTQ files are text-based files. They store nucleotide sequence and their quality scores. Then, we downloaded reference genomes. A reference genome is a typical assembly of a species' DNA sequence that acts as a foundation for individual DNA sequence alignment and interpretation. It serves as a benchmark for the annotation and comparison of novel DNA sequences. In genomics and NGS analysis, reference genomes are very crucial. We also needed to use mappers. A mapper is a tool to align DNA sequences to the reference genome. When matching reads to the reference genome, mappers take into account variables including gaps, variances, and sequencing errors. For addressing these problems, different mappers have different algorithms and approaches, and the one they choose can affect the precision and effectiveness of subsequent analyses. In our project we used BWA and Bowtie mappers. For filtering the VCF files, we

used BED files. Its main purpose is to provide a consistent and accessible representation of genomic information, including regions, intervals, and annotations. It is made up of tab-separated columns, each of which represents a distinct genetic characteristic.

We created 12 pipelines based that are distinct from each other based on their mapping methods (BWA, Bowtie), variant callers (Mutect, Strelka and Somatic-Sniper) and also base-recalibration options (with base recalibration or not). Base recalibration is a pre-processing step of DNA sequencing analysis. It helps to correct the systematic errors in base call quality scores, improving the accuracy of downstream variant calling. Base recalibration aims to identify and correct inaccuracies, ultimately enhancing the reliability of variant calls.

After running the pipelines, we used BED filtering to the obtained VCF files to get a more correct output. For that purpose, we downloaded 2 BED files. The flowchart of the steps taken can be seen at Figure 1

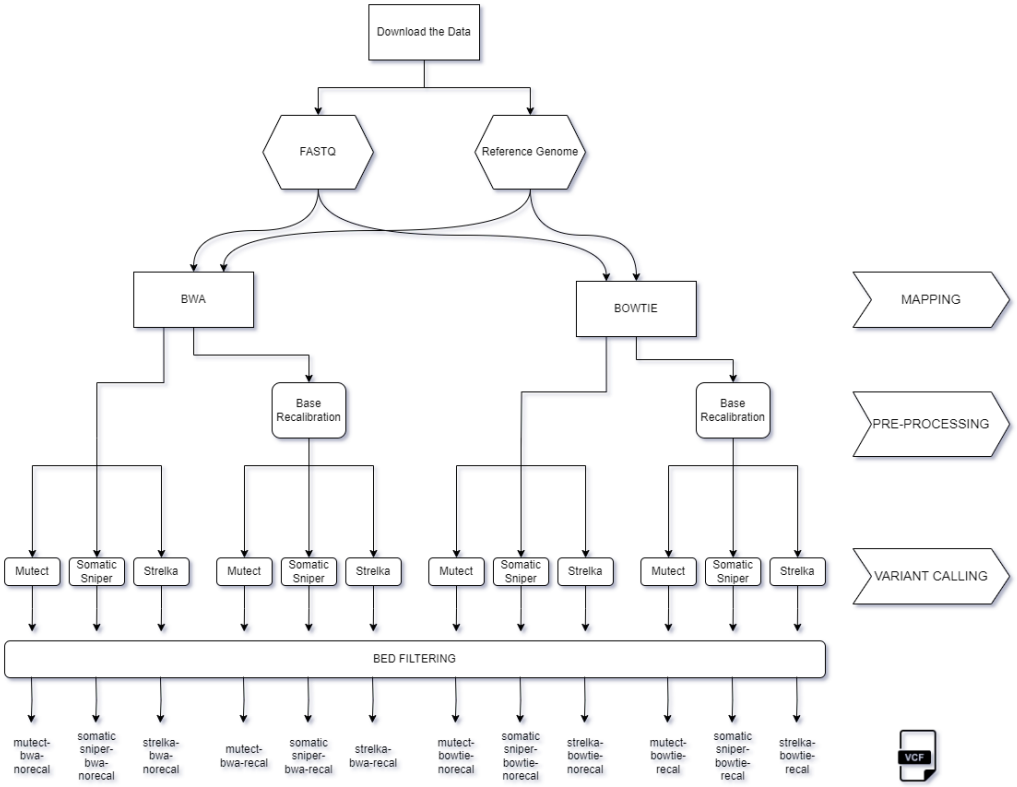

Figure 1: Flowchart of our project

After the filtering step, we calculated precision, recall, accuracy and F1 scores for each pipeline. These are the common metrics used to assess the performance of classification algorithms, including those used in variant calling from DNA sequencing data. Each of these metrics provides insights into different aspects of the algorithm's performance.

Precision is the ratio of true positive predictions to the total number of positive predictions (true positives + false positives) and it measures the accuracy of positive predictions. A high precision indicates a low rate of false positives, meaning that when the pipeline predicts a variant, it is likely to be a true variant.

Recall is the ratio of true positive predictions to the total number of actual positives (true positives + false negatives). It measures the ability of the pipeline to correctly identify all actual positive variants.

Accuracy is the ratio of correctly predicted instances to the total number of instances and provides an overall measure of how well the pipeline performs across all classes (variants and non-variants). High accuracy indicates a well-balanced performance but it may not be sufficient if the classes are imbalanced.

The F1 score is the harmonic mean of precision and recall. F1 score combines precision and recall into a single metric and provides a balance between false positives and false negatives. It is useful in scenarios with imbalanced data-sets.

For visualisation of our outputs, we used heatmaps, histograms, and PCA plots with the use of pysam, matplotlib and seaborn libraries.

3 Results

In this comprehensive analysis, twelve distinct variant calling pipelines were constructed and evaluated, employing a combination of three variant callers as SomaticSniper, Mutect, and Strelka, alongside two sophisticated mapper algorithms, BWA and Bowtie. Each of these pipelines was uniquely configured to process genomic data under two specific scenarios: one involving base recalibration and the other without recalibration. The outputs of these pipelines are Variant Caller Format (VCF) files that includes the founded variants. Then the filtered VCF files will be analysed using performance metrics and visualization technics.

3.1 Generating VCF files

Following the depicted pipeline structure, we have conducted a systematic variant calling analysis. Initially, the raw sequencing data in FASTQ format was mapped to a reference genome using BWA or Bowtie2, generating BAM files. These files were then pre-processed, which involved sorting, indexing, merging multiple BAM files if necessary, marking duplicates to identify and handle potential biases, and optionally performing base recalibration to refine the quality scores. The pre-processed BAM files were then used for variant calling with tools like Mutect, Strelka and Somatic-Sniper in the production of VCF files. These VCF files contain the identified variants from our NGS data, serving as the foundation for subsequent analysis and comparison of the different pipeline configurations.

After the generation of the VCF files using pipelines. We applied a bed filtering method using "High-Confidence_Regions_v1.2.bed" and "S07604624_Covered_human_all_v6_plus_UTR.liftover .to.hg38.bed6" files sequentially. Then we wrote a python script for further filtering. The algorithm selects the variants which have a "PASS" or "." value in "FILTER" column. The "pysam" library is used for reading the .vcf files. It is useful tool to read mapped sequence data. For the somatic-sniper outputs, We apply one more filtering using "snpfilter.pl" file from github [2].

These are the names of our final outputs generated applying previous steps:

- mutect-bowtie-norecal
- mutect-bowtie-recal

- mutect-bwa-norecal
- mutect-bwa-recal
- somaticsniper-bowtie-norecal
- somaticsniper-bowtie-recal
- somaticsniper-bwa-norecal
- somaticsniper-bwa-recal
- strelka-bowtie-norecal
- strelka-bowtie-recal
- strelka-bwa-norecal
- strelka-bwa-recal

3.2 Analysis

In the analysis part each VCF file produced by the variant calling pipelines was subjected to a comparative analysis against the reference ground truth dataset. We used the "hc_bed_filtered .recode.vcf"(high confidence VCF) file as a ground truth dataset for variants. The ground truth dataset provides a benchmark of known variant calls, which is used to assess the accuracy of the variant calls made by each pipeline. Performance metrics such as precision, recall, F1-score, and accuracy were computed to evaluate the effectiveness of the variant calling.

To calculate the performance metrics, a python script implemented. This code is reading the ground truth dataset and VCF outputs using "pysam" tool. Then we compared all the predicted variant sets with ground truth variants. Using the results, the performance metrics is calculated and printed. For a better visualization, the outputs shown in a histogram graph in Figure 3 which is created using "matplotlib" library in python. The printed values is also shown in Figure 2.

```
Metrics for mutect-bwa-recal: Precision=0.87, Recall=0.19, F1-Score=0.32, Accuracy=0.19
Metrics for mutect-bwa-norecal: Precision=0.78, Recall=0.27, F1-Score=0.4, Accuracy=0.25
Metrics for mutect-bowtie-recal: Precision=0.88, Recall=0.39, F1-Score=0.54, Accuracy=0.37
Metrics for mutect-bowtie-norecal: Precision=0.84, Recall=0.35, F1-Score=0.49, Accuracy=0.33
Metrics for strelka-bwa-recal: Precision=0.49, Recall=0.81, F1-Score=0.61, Accuracy=0.43
Metrics for strelka-bwa-norecal: Precision=0.43, Recall=0.81, F1-Score=0.56, Accuracy=0.39
Metrics for strelka-bowtie-recal: Precision=0.7, Recall=0.71, F1-Score=0.71, Accuracy=0.55
Metrics for strelka-bowtie-norecal: Precision=0.63, Recall=0.72, F1-Score=0.67, Accuracy=0.51
Metrics for somaticsniper-bwa-recal: Precision=0.37, Recall=0.74, F1-Score=0.49, Accuracy=0.33
Metrics for somaticsniper-bwa-norecal: Precision=0.3, Recall=0.75, F1-Score=0.43, Accuracy=0.27
Metrics for somaticsniper-bowtie-recal: Precision=0.45, Recall=0.67, F1-Score=0.54, Accuracy=0.37
Metrics for somaticsniper-bowtie-norecal: Precision=0.33, Recall=0.69, F1-Score=0.45, Accuracy=0.29
```

Figure 2: Performance Metrics of Pipelines

The performance metrics unveiles nuanced insights into the strengths and weaknesses of each pipeline configuration. As it can be see at Figure 3, precision is higher for Mutect compared to Strelka and Somatic Sniper. This suggests that Mutect tends to make fewer false-positive variant calls, resulting in a higher proportion of correct positive predictions. In comparison between BWA and Bowtie, Bowtie precision is higher, indicating that Bowtie tends to produce

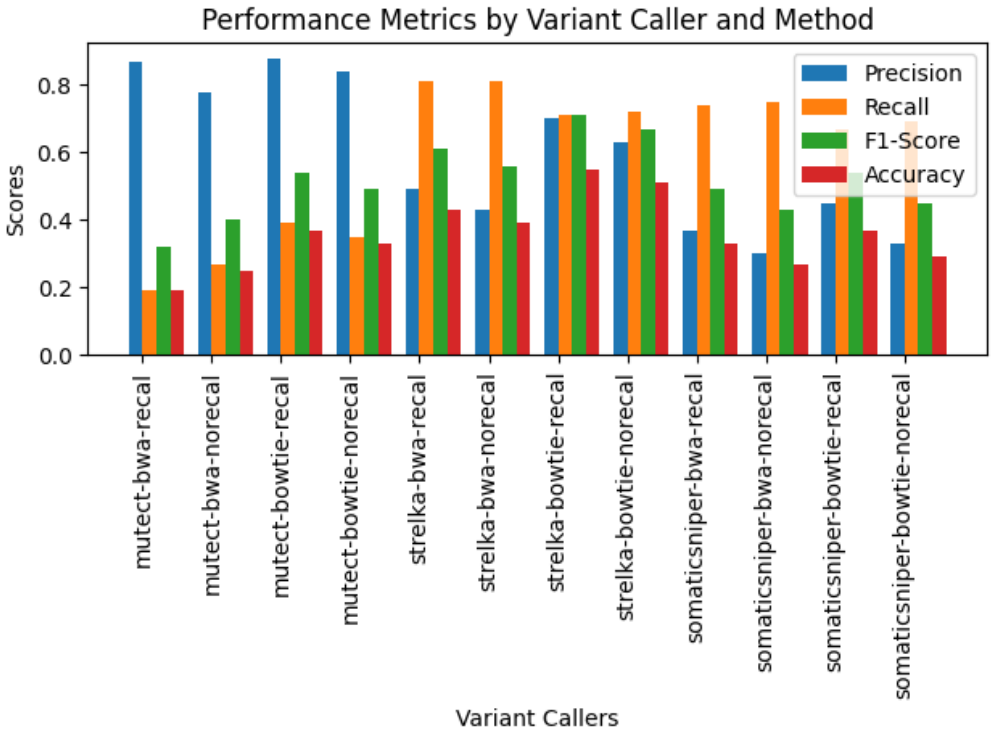

Figure 3: Histogram Graph of Performance Metrics

```
number of variants for mutect-bwa-recal : 258
number of variants for mutect-bwa-norecal : 397
number of variants for mutect-bowtie-recal : 508
number of variants for mutect-bowtie-norecal : 484
number of variants for strelka-bwa-recal : 1927
number of variants for strelka-bwa-norecal : 2204
number of variants for strelka-bowtie-recal : 1173
number of variants for strelka-bowtie-norecal : 1313
number of variants for somaticsniper-bwa-recal : 2313
number of variants for somaticsniper-bwa-norecal : 2889
number of variants for somaticsniper-bowtie-recal : 1752
number of variants for somaticsniper-bowtie-norecal : 2406
```

Figure 4: Number of variants corresponding to pipelines

fewer false positives during the mapping and variant calling processes. Also, recalibrated precision is better than not recalibrated versions. This indicates that base recalibration contributes to reducing false positives, enhancing the precision of variant calls. F1 score gives a more meaningful analysis result, since it gives a balanced output and it combines precision and recall into a single metric. F1 score seems to be highest for Strelka. To be specific, the best F1 score is obtained with strelka-bowtie-recal with 0.71 score. This suggests that Strelka, under these specific configurations, strikes an effective balance between making accurate positive predictions and capturing a substantial portion of true variants. For all outputs except for mutect-bwa's, F1 score of recalibrated versions are higher than the not recalibrated versions. Mutect and Somatic-Sniper don't have vast difference between them in terms of F1 score but Strelka most definitely has higher F1 score than both. As it can be seen, each variant caller, mapper and base-recalibration have different effects on the outputs we get.

Each variant calling algorithm (Mutect, Strelka, Somatic Sniper) employs unique methodologies, which can result in differences in precision, recall, and F1 score. The algorithms may have varying sensitivities to specific types of variants, influencing their performance. The choice of mapper algorithm (BWA vs. Bowtie) contributes to differences in precision. The mapping algorithm affects how reads are aligned to the reference genome, impacting the accuracy of variant calling. The notable improvement in precision and F1 score with base recalibration suggests that this step is effective in reducing systematic errors, resulting in more accurate variant calls.

The number of variants corresponding to the pipelines can be seen at Figure 4. As it can be seen, number of variants is lowest for mutect and highest for somatic sniper which may indicate that mutect is more selective in terms of the selection of the variants.

The heatmap visualization serves as a critical analysis tool, representing the comparative landscape of the 12 VCF files generated by the different variant calling pipelines. By utilizing Jaccard distance as a measure of dissimilarity, the heatmap provides an intuitive graphical representation of how closely related each pair of VCF files is, based on their variant calls. The color intensity in each cell of the heatmap corresponds to the degree of overlap between the variant sets: blue colors signify higher similarity (lower Jaccard distance), while red colors indicate less similarity (higher Jaccard distance). This visual aid effectively consolidates the complex variant data into an accessible format, allowing for an quick assessment of the performance and concordance of the various pipeline configurations tested in the study. It highlights the nuances of the variant calling results, such as which pipelines tend to agree and which ones diverge, providing a foundation for further qualitative and quantitative evaluation.

In the implementation, the resulting Jaccard distances of VCF files were organized into a matrix, where each cell represented the distance between a pair of VCF files. This matrix served as the input for the heatmap visualization, rendered using the Seaborn library. This is the output of our python implementation in Figure 5.

As seen in heatmap of 12 distinct variant outputs, the pipelines using Mutect as the variant caller, generally show higher similarity with each other, as indicated by the cooler colors in the corresponding rows and columns. We can understand that the mutect variant caller outputs has a higher precision value, because it is not differentiate when different mapper algorithms applied and used recalibration or not. The pipeline using Strelka as the variant caller has a better similarity when compared to Somatic-sniper outputs, so it can have a higher accuracy predicting the variants in sequence. Only in the inference we get from the diagram, Somatic-sniper variant caller can not be a proper choice for this sequence, because it is significantly differentiate when different mapper algorithms applied and used recalibration or not. Also it have lowest similarity values when compared to other variant callers. When the Jaccard distance values examined, the recalibration method has not a significant effect when compared to other pipelines.

As seen in Figure 6, we improved the heatmap plot adding dendograms to see the similarities of the pipelines. Dendrograms, which are tree-like diagrams used to illustrate the arrangement of the clusters produced by hierarchical clustering, provide a clear and intuitive representation of the relationships and similarities between different pipelines. When integrated with the heatmap, they offer a comprehensive overview of the data structure at a glance.

To compare in terms of variants, The heatmap and dendogram plottings of pipelines constructed. The inclusion of dendrograms aids in identifying groups of pipelines with similar

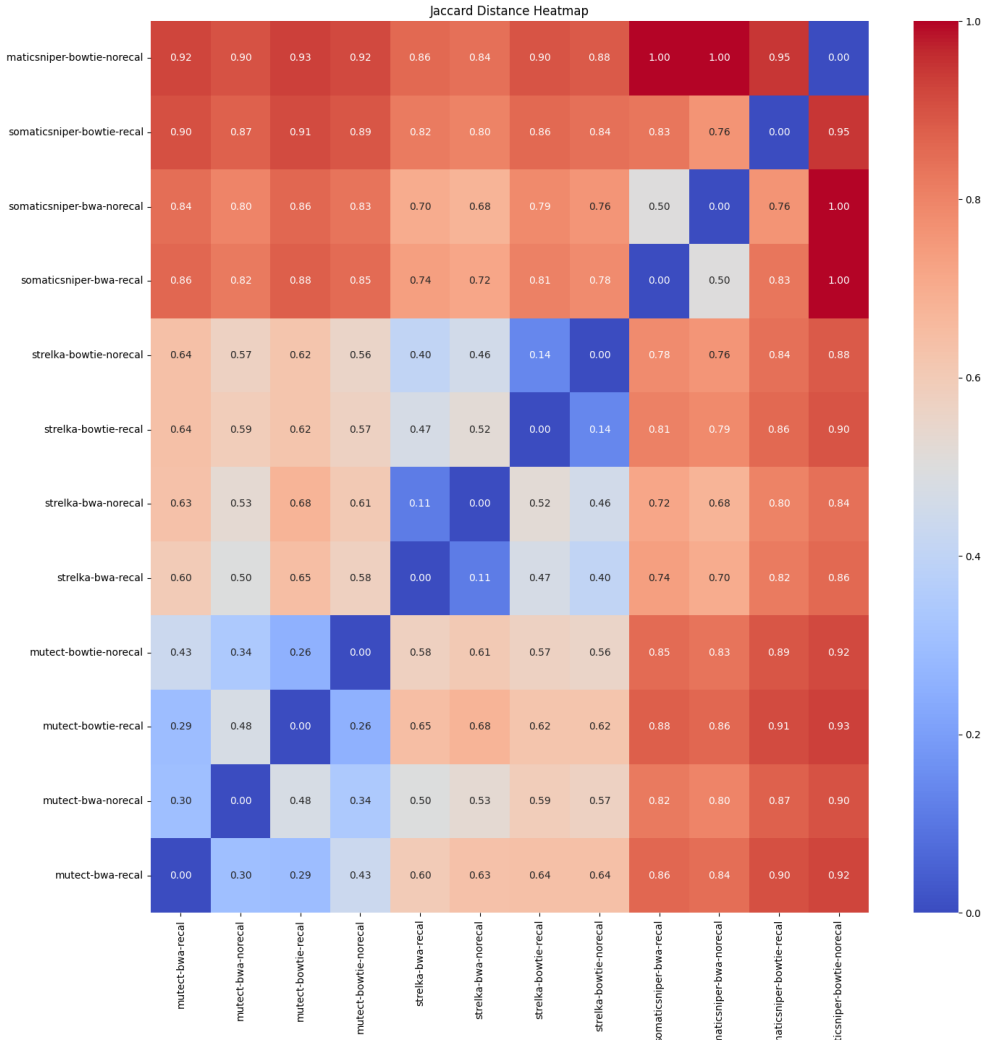

Figure 5: Heatmap of Pipelines

variant profiles, highlighting common patterns or discrepancies that might not be immediately apparent from the heatmap alone. On the y-axis, the dendrogram groups pipelines that detect similar sets of variants, which can be crucial for understanding the performance and specificity of different variant calling methods. Meanwhile, the dendrogram on the x-axis clusters variants based on their presence across multiple pipelines, revealing potential recurrent or unique variants.

For constructing the Principal Component Analysis (PCA) on the VCF files, we initiated the process by extracting variant data from each file. The variant information—comprising chromosome, position, reference, and alternate alleles—was aggregated from the 12 distinct VCF files generated by the pipelines using BWA and Bowtie mappers, with and without base recalibration, and employing variant callers including SomaticSniper, Mutect, and Strelka. Each VCF file was translated into a binary vector, where each element represented the presence or absence of a specific variant across all files, creating a binary matrix. This matrix served as the input for PCA, allowing us to reduce the dimensionality of the data and identify the principal components that captured the most significant variance within the dataset. By fitting the PCA model to this binary matrix and selecting the top three or two principal components,

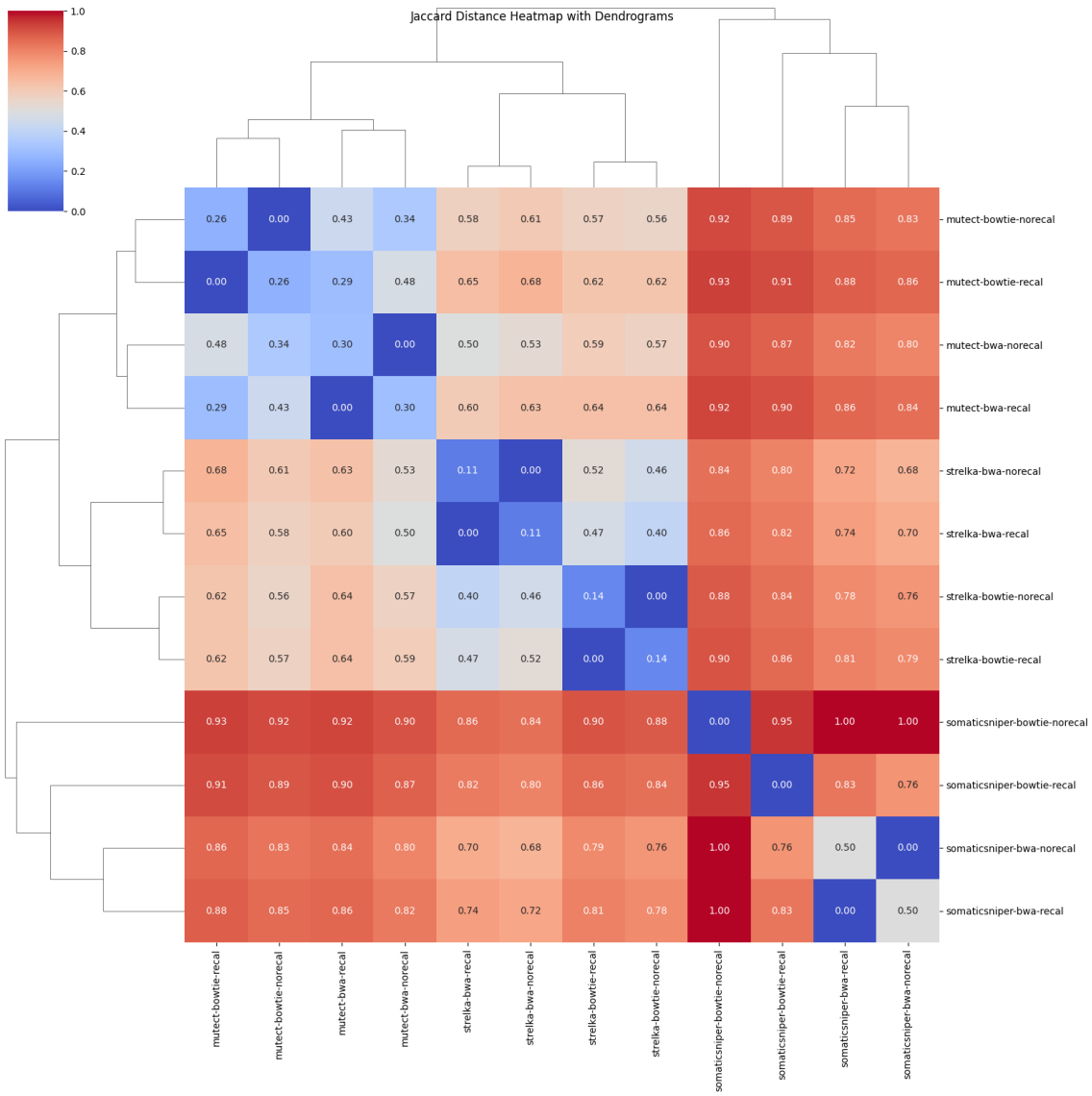

Figure 6: Heatmap of Pipelines with Dendogram

we were able to visualize the high-dimensional variant data in a three-dimensional and two-dimentional space. The resulting scatter plot offered a clear view of the clustering patterns and relationships between the different pipeline outputs, facilitating an intuitive understanding of their comparative performance in variant calling. We used scikit-learn library to apply the PCA algorithm. These are the outputs of our PCA implementation in Figure 8 and Figure 9.

The PCA outputs presented in the plots visualizes the variation in VCF files generated by different combinations of variant callers and mappers with or without the recalibration process. Within each variant caller group, the recalibrated (recal) and non-recalibrated (norecal) pipelines do not show a significant separation, suggesting that recalibration does not have a major impact on the principal components derived from these VCF files. The SomaticSniper (blue) and Mutect (red) pipelines are closer to each other on the plot, suggesting more similarity in their variant calling outputs as compared to Strelka. The SomaticSniper-Bowtie pipeline with recalibration appears to be an outlier, displaying a unique variation pattern compared to the rest of the pipelines.

Figure 7: Heatmap of Pipelines with Variants

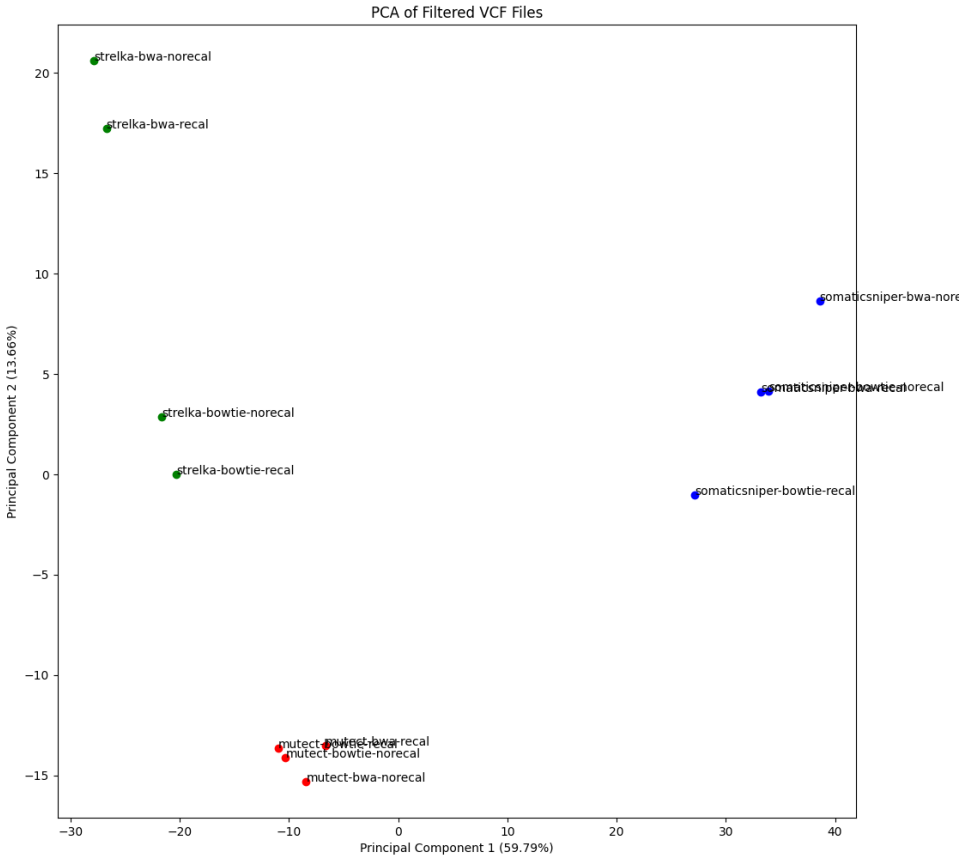

Figure 8: 2D PCA of Pipelines

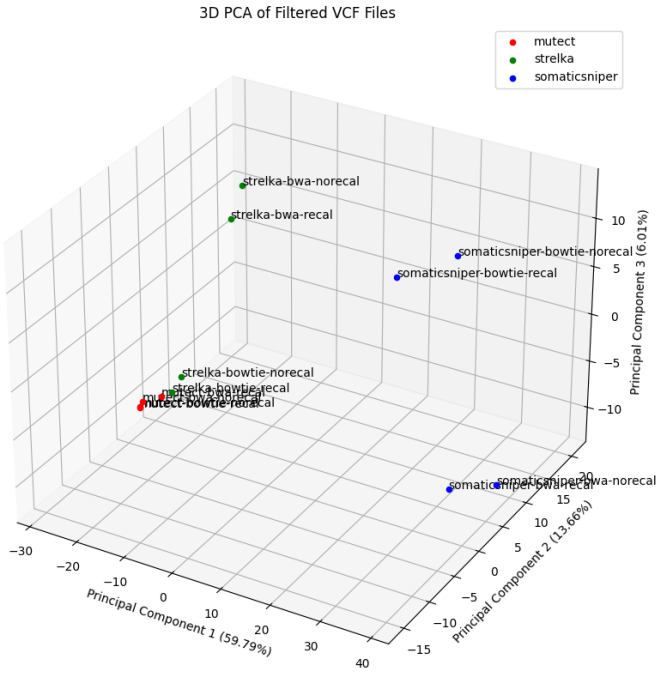

Figure 9: 3D PCA of Pipelines

4 Discussion

In this study,we delve into the interpretations of the variant calling pipeline analysis outcomes, informed by the performance metrics comparison with a ground truth dataset, heatmap visualizations, and PCA results. The metrics revealed that the pipelines employing Mutect and Strelka as variant callers with BWA for mapping and recalibration tended to have higher precision, indicating a strong ability to correctly identify true variants. However, these pipelines showed varied recall values, reflecting a disparity in their capacity to capture all true variants present in the ground truth data. Notably, Strelka pipelines, particularly with recalibration, demonstrated a balanced performance with a relatively higher F1-score, suggesting an effective compromise between precision and recall.

The heatmap analysis, underpinned by Jaccard distance calculations, furnished additional insights, illustrating that while some pipelines shared substantial similarities in variant calling, others diverged significantly, as evidenced by the variance in color intensities. This divergence indicates the inherent differences in algorithmic approaches to variant calling and underscores the potential for complementary insights when multiple pipelines are considered in tandem.

PCA visualization further elucidated the distinct patterns in the data, with the separation of pipelines along the principal components underscoring the variability introduced by different combinations of variant callers, mappers, and the recalibration process. The grouping of pipelines with similar variant callers, regardless of the mapper used, highlighted the dominant influence of the variant calling algorithm in the variance of the outputs.

For future research, it would be prudent to explore the underlying causes of the observed

discrepancies in performance metrics. This could involve a deeper investigation into the algorithmic nuances of the variant callers and the biological implications of the variants they preferentially detect or overlook. Additionally, examining the impact of recalibration in more depth could yield insights into the conditions under which this process most effectively enhances the accuracy of variant calls. The integration of more diverse datasets, including those with different genetic backgrounds and variant profiles, could refine the understanding of pipeline performance in varied genomic contexts. Lastly, expanding the analysis to include additional variant callers and mappers, as well as exploring alternative metrics of comparison, could provide a more comprehensive picture of the landscape of NGS data processing tools.

References

[1] Tiago Antao. *Bioinformatics with Python Cookbook*. Packt Publishing Ltd, 2022.

[2] *"somatic-sniper/src/scripts at master·genome/somatic-sniper,"* GitHub. https://github.com/genome/somatic-sniper/tree/master/src/scripts (accessed Dec. 20, 2023).