Title: Variant Calling from SARS-CoV-2 Nanopore Sequencing Data Using Cloud Computing

Team members: Poojitha Kolli, Swetha Yadavalli, Yesasvi Sai Nandigam

Introduction

Understanding genetic variations is fundamental to advancing genomics and healthcare. Variant calling, a key bioinformatics process, identifies genetic differences such as single nucleotide polymorphisms (SNPs), insertions and deletions (Indels), and structural variations. These insights are critical for studying genomic diversity, monitoring viral evolution, identifying pathogen mutations like those in SARS-CoV-2, and driving the development of diagnostics and therapeutics. High-throughput sequencing technologies have revolutionized genomics by enabling the rapid generation of large-scale data, but the computational requirements for processing and analyzing these datasets pose significant challenges.

Cloud computing has emerged as a transformative solution for modern bioinformatics applications, addressing the growing demand for scalable and flexible computational resources. Platforms like Amazon Web Services (AWS) provide researchers with the ability to dynamically allocate resources based on workload demands, ensuring efficiency and cost-effectiveness. Beyond resource scalability, cloud platforms foster global collaboration by offering standardized and reproducible environments accessible to geographically distributed teams. These features make cloud computing an indispensable tool for handling large-scale genomic analyses, including SARS-CoV-2 variant detection, despite persistent limitations like data transfer costs and latency.

Problem description

SARS-CoV-2 nanopore sequencing generates complex and voluminous data that require efficient and accurate variant calling to support genomic surveillance and public health initiatives. Current workflows face considerable challenges, particularly in the alignment of reads, detection of mutations, and filtering of results. These steps are often complex and computationally demanding, which can hinder the efficiency and accuracy of variant calling. These challenges are amplified when dealing with large datasets or multiple samples, where timely processing and scalability are critical but often unattainable using traditional high-performance computing (HPC) systems.

Traditional high-performance computing (HPC) systems, while powerful for processing large datasets, are not ideal for the dynamic and fluctuating computational demands required in genomic research, particularly when dealing with sequencing data. These systems require pre-allocation of resources, incur high operational costs, and often suffer from long wait times for job completion. Furthermore, their rigid infrastructure can create bottlenecks in workflows, especially when handling massive datasets, limiting scalability and collaboration.

In contrast, cloud-based platforms offer a more flexible and cost-effective solution. By providing on-demand access to scalable resources, cloud platforms eliminate the need for infrastructure management and allow researchers to dynamically adjust computational power based on workload requirements. This reduces waiting times, improves efficiency, and lowers costs, as resources are used only when needed.

Moreover, cloud environments can be accessed from anywhere, fostering collaboration and enabling faster, more reproducible results in genomic surveillance and public health initiatives.
To address these challenges, this project proposes a cloud-based platform designed to process SARS-CoV-2 nanopore sequencing data. By leveraging Amazon EMR with Apache Spark, the platform aims to deliver dynamic resource allocation, efficient variant calling workflows, and seamless scalability.

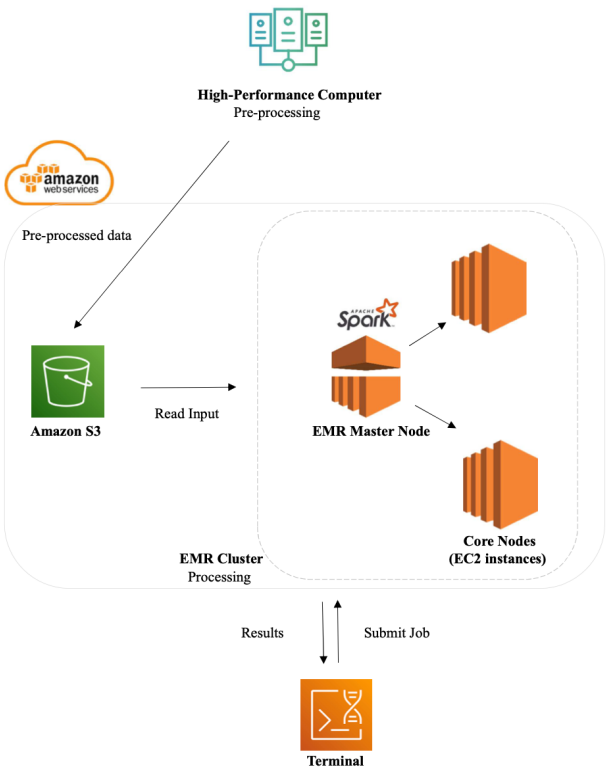
Data description

The analysis in this project is based on 15 SARS-CoV-2 Nanopore sequencing samples, each approximately 30 MB in size (PRJNA750736). These samples were chosen for their relevance in identifying genetic variations in the SARS-CoV-2 genome. The data were preprocessed and stored in the cloud for efficient access and analysis.

To process and analyze this data, Amazon Web Services (AWS) was used as the cloud provider, offering a reliable and scalable solution for genomic data workflows. Specifically, Amazon Elastic MapReduce (EMR), combined with Apache Spark, was employed for distributed data processing. Spark's ability to distribute tasks across multiple nodes allows for fast and scalable analysis of genomic datasets.

Methodology

This flowchart illustrates the project workflow, as outlined below:



Preprocessing on HPC:

The raw sequencing data was preprocessed on a High-Performance Computer (HPC) to ensure its quality and suitability for downstream analysis. The preprocessing workflow included quality assessment using the FASTQC tool, removal of low-quality reads and adapters, and trimming with Trim Galore.

Data Upload to AWS S3:

An AWS free-tier account was created, and an S3 bucket was set up to store the trimmed files from preprocessing, along with the reference genome needed for alignment. Utilizing S3 enabled efficient data management and provided easy accessibility for cloud processing, ensuring seamless integration with the EMR cluster in the computational workflow.

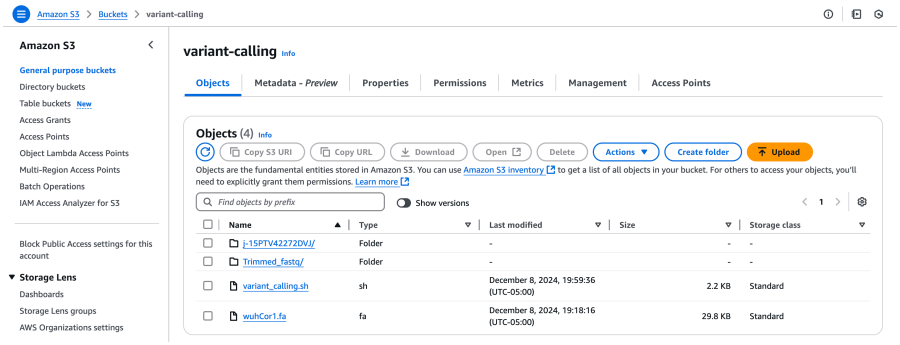


Figure 1: The Amazon S3 bucket "variant-calling" was created to store the pre-processed files, reference genome and the variant calling script.

EMR Cluster Setup:

An Elastic MapReduce (EMR) cluster was configured to process genomic data efficiently using distributed computing with Apache Spark. The cluster was set up with a master node for task management and two core nodes (EC2 instances) for parallel processing.

To ensure secure and seamless operation, two key IAM roles were defined:

1. **Service Role for EMR (AmazonElasticMapReduceRole):**
   This role allowed the EMR service to manage cluster resources and interact with other AWS services, such as S3 for data storage.
2. **Instance Profile Role for EC2 (AmazonElasticMapReduceforEC2Role):**
   This role provided EC2 instances within the cluster with the necessary permissions to access S3, manage logs, and perform related tasks.

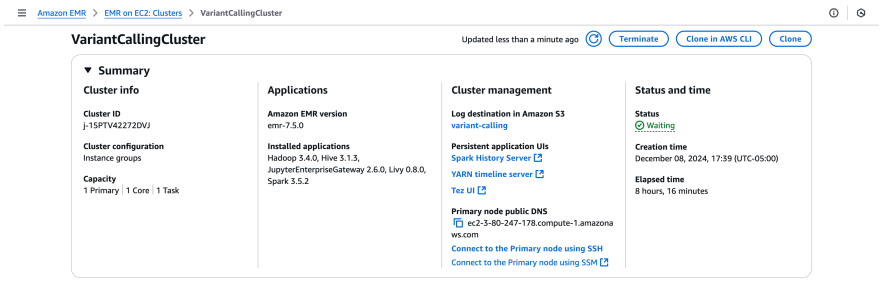


Figure 2: An Amazon EMR cluster named "VariantCallingCluster" was created, connected to the S3 bucket, and includes a primary node public DNS, enabling terminal access to the cluster.

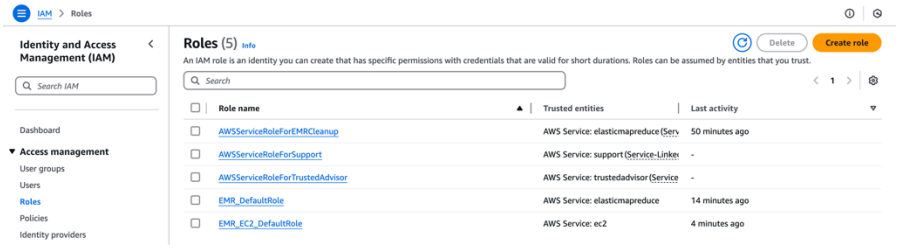


Figure 3: Two key IAM roles, the Service Role for EMR (EMR_DefaultRole) and the Instance Profile Role for EC2 (EMR_EC2_DefaultRole) enabled secure EMR cluster management and EC2 permissions for accessing S3 and managing logs.

Variant Calling Tools Installation

The tools required for the variant calling pipeline were downloaded and installed on the EMR cluster by logging into the master node via SSH through the terminal. Minimap2, Samtools, and LoFreq were set up to enable alignment, data processing, and variant detection.

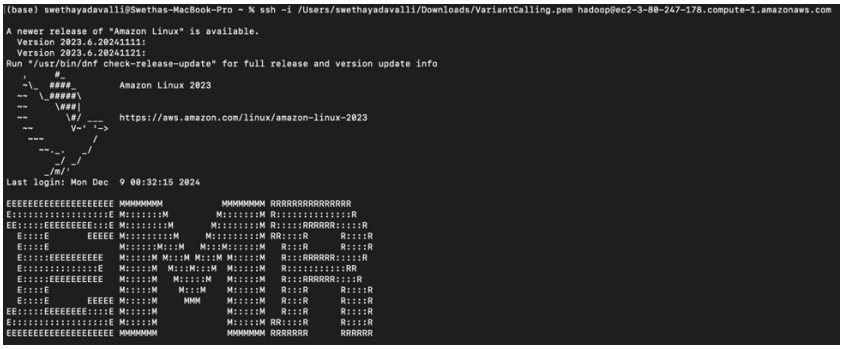


Figure 4: Logged into the master node of the EMR cluster via SSH through the terminal

Variant calling workflow:

The variant calling workflow was implemented using a script that included steps for alignment, conversion of SAM to BAM files, and final variant detection. Minimap2 was employed to align sequencing reads to the reference genome with speed and accuracy. Samtools handled the processing of alignment data by converting it into BAM files for efficient management. LoFreq was then used to detect SNPs and Indels with high precision. The job was submitted through the terminal interface, enabling communication with the EMR cluster for executing the pipeline. Once the analysis was completed, the terminal was also used to retrieve the processed results.

Results

The cloud-based platform significantly outperformed traditional HPC systems in processing time. After preprocessing, the pipeline is completed in 1 hour on the cloud, compared to 2 hours on HPC. This highlights the cloud's superior scalability and dynamic resource allocation, which reduces processing time and improves efficiency for large genomic datasets like SARS-CoV-2 nanopore sequencing.

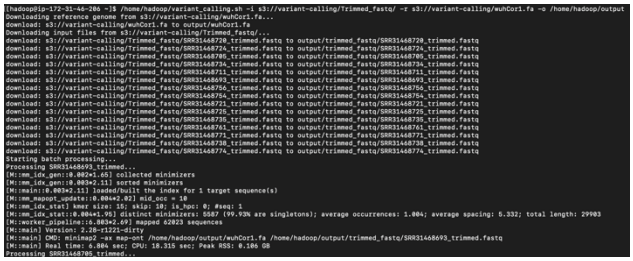


Figure 5: Execution of Variant calling pipeline from a terminal connected to the EMR cluster

```
[hadoop@ip-172-31-46-206 ~]$ ls
htslib  htslib-1.17  htslib-1.17.tar.bz2  lofreq_star-2.1.5_linux-x86-64  lofreq_star-2.1.5_linux-x86-64.tgz  minimap2  output  samtools-1.17  samtools-1.17.tar.bz2  variant_calling.sh
[hadoop@ip-172-31-46-206 ~]$ cd output
[hadoop@ip-172-31-46-206 output]$ ls
BAM  VCF  align  trimmed_fastq  wuhCor1.fa  wuhCor1.fa.fai
```
Figure 6: Outputs are stored in the output folder created in the terminal session

```
[hadoop@ip-172-31-46-206 VCF]$ cat SRR31468693_trimmed.vcf
##fileformat=VCFv4.0
##fileDate=20241209
##source=lofreq call -f /home/hadoop/output/wuhCor1.fa -o /home/hadoop/output/VCF/SRR31468693_trimmed.vcf /home/hadoop/output/BAM/SRR31468693_trimmed_sorted.bam
##reference=/home/hadoop/output/wuhCor1.fa
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw Depth">
##INFO=<ID=AF,Number=1,Type=Float,Description="Allele Frequency">
##INFO=<ID=SB,Number=1,Type=Integer,Description="Phred-scaled strand bias at this position">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Counts for ref-forward bases, ref-reverse, alt-forward and alt-reverse bases">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=CONSVAR,Number=0,Type=Flag,Description="Indicates that the variant is a consensus variant (as opposed to a low frequency variant).">
##INFO=<ID=HRUN,Number=1,Type=Integer,Description="Homopolymer length to the right of report indel position">
##FILTER=<ID=min_dp_10,Description="Minimum Coverage 10">
##FILTER=<ID=sb_fdr,Description="Strand-Bias Multiple Testing Correction: fdr corr. pvalue > 0.001000">
##FILTER=<ID=min_snvqual_69,Description="Minimum SNV Quality (Phred) 69">
##FILTER=<ID=min_indelqual_20,Description="Minimum Indel Quality (Phred) 20">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO
NC_045512v2     241     .       C       T       42031   PASS    DP=1567;AF=0.890874;SB=0;DP4=28,0,1425,0
NC_045512v2     670     .       T       G       25259   PASS    DP=1446;AF=0.857538;SB=0;DP4=100,0,1270,0
NC_045512v2     897     .       C       A       22264   PASS    DP=1037;AF=0.880424;SB=0;DP4=16,0,950,0
NC_045512v2     1435    .       A       G       25266   PASS    DP=1128;AF=0.939716;SB=0;DP4=40,0,1076,0
NC_045512v2     2198    .       G       A       290     PASS    DP=1536;AF=0.132161;SB=0;DP4=1225,0,265,0
NC_045512v2     2790    .       C       T       22422   PASS    DP=1087;AF=0.848206;SB=0;DP4=13,0,940,0
NC_045512v2     3037    .       C       T       24307   PASS    DP=1065;AF=0.845070;SB=0;DP4=9,0,902,0
NC_045512v2     3431    .       G       T       7394    PASS    DP=271;AF=0.900369;SB=0;DP4=1,0,246,0
NC_045512v2     3565    .       T       C       5123    PASS    DP=257;AF=0.898833;SB=0;DP4=9,0,238,0
NC_045512v2     3714    .       C       T       5577    PASS    DP=232;AF=0.818965;SB=0;DP4=4,0,194,0
NC_045512v2     4184    .       G       A       2825    PASS    DP=156;AF=0.903846;SB=0;DP4=5,0,145,0
NC_045512v2     4321    .       C       T       13503   PASS    DP=1839;AF=0.649266;SB=0;DP4=339,0,1363,0
NC_045512v2     5182    .       T       C       30378   PASS    DP=1592;AF=0.768216;SB=0;DP4=39,0,1232,0
NC_045512v2     5736    .       C       T       1309    PASS    DP=707;AF=0.308345;SB=0;DP4=423,0,232,0
NC_045512v2     6078    .       C       T       333     PASS    DP=679;AF=0.170839;SB=0;DP4=534,0,125,0
NC_045512v2     6183    .       A       G       8045    PASS    DP=530;AF=0.839623;SB=0;DP4=41,0,465,0
```
Figure 7: Viewing an output file

Discussion

Modern genomic analysis frameworks are highly adaptable, offering accessibility, reproducibility, and flexibility through the use of open-source tools and cloud platforms. These frameworks support customizable workflows that can accommodate a range of applications, including transcriptomics,

epigenomics, and metagenomics, by incorporating various data types such as RNA-Seq and ChIP-Seq. This adaptability enables detailed studies of regulatory mechanisms and disease-associated variants, making it easier to perform large-scale genomic analyses.

However, these frameworks also come with some limitations. Uploading and downloading large genomic datasets to and from cloud storage can be expensive and slow, particularly when dealing with datasets spanning terabytes. Additionally, transferring large volumes of data over long distances can introduce network latency and bandwidth limitations, further slowing down the workflow. Furthermore, cloud platforms, while offering scalability, can lead to unpredictable costs if resource usage is not carefully managed. These limitations underscore the importance of optimizing data transfer processes and managing cloud resources to ensure efficient and cost-effective genomic analysis.

References

1. *Variant Calling - CD Genomics*. (n.d.). https://www.cd-genomics.com/variant-calling.html

2. Al-Ars, Z., Wang, S., & Mushtaq, H. (2020). SparkRA: Enabling Big Data Scalability for the GATK RNA-seq Pipeline with Apache Spark. Genes, 11(1), 53. https://doi.org/10.3390/genes11010053

3. Sharma, R. (2023, May 3). An overview of variant calling and analysis in NGS data. Basepair. https://www.basepairtech.com/blog/variant-calling-in-genomics/#:~:text=Variant%20calling%20is%20an%20important,differences%20between%20individuals%20and%20populations.

4. Decap, D., De Schaetzen Van Brienen, L., Larmuseau, M., Costanza, P., Herzeel, C., Wuyts, R., Marchal, K., & Fostier, J. (2022). Halvade somatic: Somatic variant calling with Apache Spark. GigaScience, 11. https://doi.org/10.1093/gigascience/giab094

Appendix 1:

Poojitha Kolli – Pre-processed the raw files in HPC, implemented variant calling pipeline (Minimap2, Samtools, LoFreq), report writing (results, discussion)

Swetha Yadavalli - Tools installation, Configured EMR cluster and managed IAM roles, report writing (Data description and Workflow)

Yesasvi Sai Nandigam - Searched through and acquired datasets, Set up S3 bucket for data storage, , report writing (Introduction and Problem Description)