SnakeLines Workflow for SARS-CoV-2 Variant Detection from Next-Generation Sequencing Reads

Adrián Goga[1,2], Miroslav Böhmer[2,3,4,5], Rastislav Hekel[3], Werner Krampl[2,3,5], Bronislava Brejová[1], Tomáš Vinař[1], Jaroslav Budiš[2,3,6], and Tomáš Szemes[2,3,5]

[1] Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Slovakia,
[2] Comenius University Science Park in Bratislava, Slovakia,
[3] Geneton Ltd., Bratislava, Slovakia,
[4] Slovak Centre of Scientific and Technical Information, Slovakia,
[5] Faculty of Natural Sciences, Comenius University in Bratislava, Slovakia
[6] Slovak Centre of Scientific and Technical Information, Slovakia

Abstract: The ongoing SARS-CoV-2 pandemic, which emerged in December 2019, revolutionized genomic surveillance, leading to new means of tracking viral spread and monitoring genetic changes in their genomes over time. One of the key sequencing methods used during the pandemic is based on massively parallel short read sequencing based on Illumina technology. In this work, we present a highly scalable and easily deployable computational pipeline for the analysis of Illumina sequencing data, which is used in Slovak SARS-CoV-2 genomic surveillance efforts. We discuss several issues that arose during the pipeline design, and which could both provide useful insight into the analysis processes and serve as a guideline for optimized future outbreak surveillance projects.

1 Introduction

The Severe Acute Respiratory Syndrome Coronavirus type 2 (SARS-CoV-2) pandemic first emerging in Wuhan in December 2019, also erupting in Slovakia in March of 2020, led to a revolution in genomic surveillance. Multiple DNA sequencing library preparation protocols and data analysis pipelines have been utilized to keep track of the rapidly evolving virus, for example ARTIC [20], SARS-CoV-2 RECoVERY [6], V-pipe [25], etc. Their use resulted in gathering millions of genomic sequences in the GISAID database [28].

We present a computational pipeline, which was crucial for the analysis of Illumina sequencing data of SARS-CoV-2 genomes in Slovakia. The pipeline was developed using the SnakeLines framework [4], which is built upon the widely used Snakemake workflow engine [14]. The flexibility of this framework allowed us to deploy our solution within national computational infrastructures supporting sequencing laboratories of Comenius University Science Park in Bratislava and the Public Health Authority of the Slovak Republic. Up to now (from mid-February to late July of 2021), we have successfully sequenced, analysed and released 3055 viral samples with sufficient coverage (at least 3bp across more than 90% of the genome).

The rest of this paper is organized as follows. In section 2, we give a brief overview of the DNA sequencing process and data analysis with focus on variant detection of the SARS-CoV-2. In section 3, we describe the SnakeLines-based variant detection pipeline we designed for the SARS-CoV-2. Section 4 is focused on discussing several selected problems that occurred during the pipeline design. Finally, in section 5 we conclude our findings, propose several recommendations based on our experience, and outline directions for future work.

2 Background - Sequencing and Data Analysis

The reference genome of the SARS-CoV-2 has 29 903 base pairs (bp) and was sequenced in the early days of the epidemic [31]. The goal of sequencing of additional samples is to identify differences from this reference sequence, called genomic variants. Continuous monitoring of viral genomes is crucial for the early detection of new mutants that may lead to more severe virulence characteristics, such as the rate of infection.

A typical analysis of genomic data can be divided into three consecutive steps. At first, the input genomic material is fragmented and digitalized using sequencers. The fragments are then compared to a related genome to identify genomic variations. Finally, the effect of detected variations is estimated.

SARS-CoV-2 genomes are typically sequenced from samples collected for RT-qPCR diagnostic tests. The next step is the library preparation process, which we have done using the Illumina COVIDSeq Test. The Illumina COVIDSeq Test leverages a modified version of the validated, publicly available ARTIC multiplex PCR protocol. It involves a two-step multiplex PCR approach in which the whole genome is amplified using 98 amplicon primers. Multiplexing (or barcoding) is often used to enable simultaneous sequencing of multiple samples by ligating unique identifier sequences, called barcodes, to the reads that belong to a particular sample.

The amplified DNA fragments are then subjected to a sequencing process according to the technology being used, which in our case is the Illumina Next Generation Sequencing (Illumina NGS). Illumina NGS sequences a predefined number of bases from both ends of each DNA fragment (we have used 2 × 36bp and 2 × 74bp read lengths) to form 'paired-end' reads (see the left of Fig. 1). The Illumina NGS reads, albeit much shorter than those of the Third Generation Sequencing technologies (for example the Oxford Nanopore Technologies), are significantly more accurate on the base level, which makes them generally more suitable for calling lower frequency variants that may indicate co-infection with multiple strains in a patient, evolution of the virus in a patient, or laboratory contamination [23, 27].

Even though the fragments are not sequenced completely, the length of the unsequenced middle part is known, which proves useful in locating the read's position of origin. The output of the sequencing process are the signal intensities of individual dNTPs (deoxynucleoside triphosphates) measured through laser excitation and imaging [11], from which the nucleotide sequences are extracted in the base calling procedure. The base called reads occasionally contain errors that complicate further analyses. The measure of these errors is reflected in the quality scores (Q-scores), which are generated for each base and represent the probability of erroneous base calls [10]. Upon obtaining the base called reads, demultiplexing - the inverse step to multiplexing can be executed so that the reads are assigned to the sequenced samples on the basis of their barcodes.

The output of the sequencing process is then subjected to a series of steps according to the type of analysis being performed. At the top level, we distinguish between reference-based and reference-free analysis on the basis of whether it constructs the sequenced genome by comparing it to a known reference genome or it does it using solely the sequenced data.

In the reference-based analysis, the sequenced reads are directly compared to the reference genome to identify differences in their sequences, called genomic variants. From a higher perspective, the individual steps of the reference-based analysis can be described as follows:

1. Reads preprocessing
2. Mapping to a reference
3. Variant calling
4. Consensus construction
5. Generation of summary reports

Step 1 comprises a set of procedures that clear the reads of any laboratory artifacts that could degrade the results of a downstream analysis. These procedures include, e.g. the removal of low quality parts of the reads, elimination of foreign DNA sources (e.g. human), removal of the fragments duplicated by the PCR process in case of the Illumina sequencing technology, etc.

The genomic regions from which the reads originate are located by performing alignments to the reference sequence in step 2. This is conducted using one of the reference-mapping tools such as the widely used Bowtie2 or BWA [15, 17]. The data describing the mappings (usually in the SAM format) is subsequently sorted by the reference position and indexed for quick access using the SAMtools program [18]. Further postprocessing is possible, e.g. discarding the reads that do not meet the mapping quality threshold, etc.

The step 3 - variant calling - is the identification of variations that are present in the sequenced samples. These variations of interest consist mainly of single-nucleotide polymorphisms (SNPs) and small indels, and are detected using tools like FreeBayes, GATK, or BCFtools [8, 24, 18]. The goal of these tools is to distinguish sequencing errors in reads from real variants based on the number of reads containing the variant, their quality, read mapping quality and other data. The identified variants are incorporated into the consensus sequence of the sequenced sample in the next step 4. The positions in which the sequencing coverage (i.e. the number of mapped reads covering the position) is less than a predefined threshold get marked by the 'N' symbol, which stands for any base. This consensus can be then used to identify the most likely phylogenetic lineage by matching it to a database of known lineages using, for example, the Pangolin tool [22]. The steps 2-4 are depicted in Figure 1.

Finally, comprehensible reports in the PDF/HTML format describing the summary statistics and plotted graphs are generated for each individual step of the analysis. These reports are used to monitor the quality of a given step or to detect a potential problem with the analysed sample, caused by laboratory or computational processing. This is usually done using a combination of specialized reporting tools, such as FastQC, MultiQC or Qualimap [3, 7, 21].

The reference-free approach shares the first and last step with the reference-based one, but the middle step consists of a genome assembly, in which the reads are joined into long sequences called contigs. This is done by observing the extent of overlaps in individual read pairs. The contigs can be further assembled into 'scaffolds' in which the distances between contig pairs are known. The tools used for the assembly include Spades, Unicycler and Megahit [1, 32, 16], with the summary reports of the assembly quality reported by Quast [9] and the assembled graph visualized by Bandage [33]. The reference-free approach is suitable for analyses of viruses without a sufficiently similar reference sequence. Furthermore, it is able to detect larger structural changes, e.g., longer insertions.

The MALVIRUS web application demonstrates a novel approach to reference-free variant calling, which uses a

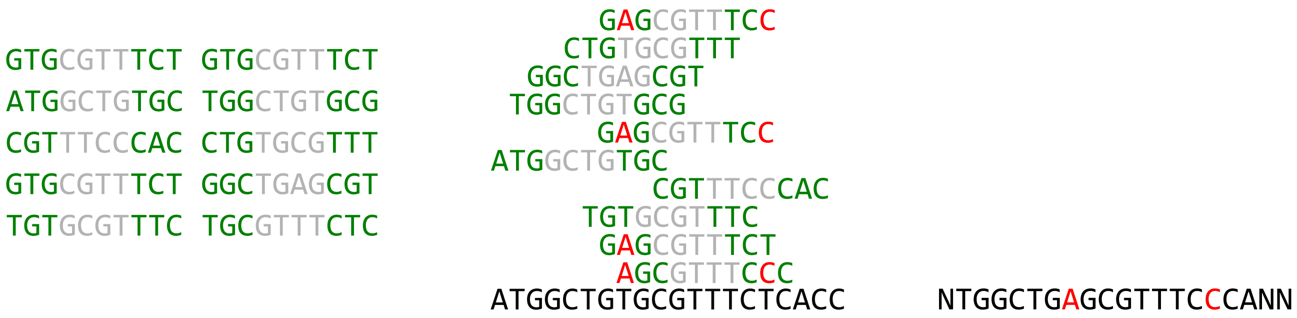

Figure 1: A schematic visualization of the Illumina NGS data analysis. Left: a set of fragments, each having 3 bp sequenced from its ends. Middle: the reads mapped to the reference sequence with the identified variants highlighted (red). Right: the consensus sequence of this example where the positions with coverage lower than 2 were marked by 'N'.

catalog of known variants from the viral population (built utilizing databases such as GISAID) to genotype newly-sequenced samples in an alignment-free fashion [5].

3 Implementation of Our Pipeline

3.1 SnakeLines Workflow Framework

Analyses of DNA sequencing datasets typically consist of a number of steps performed by individual tools in a particular order, which is very time demanding when executing each tool manually by an operator. This naturally leads to development of automated computational pipelines, many of which also support the isolation of used tools in individual environments or parallel execution of independent computations. Another desirable property of computational pipelines is the simplification of analysis reproducibility between different data analysis centers.

An increasingly popular choice for pipeline building in bioinformatics is the Snakemake [14] workflow engine. The development of a pipeline in this engine consists of defining a set of basic building blocks called rules in a Python-based language [14]. Each rule defines commands for transforming a set of inputs into a set of outputs, where both inputs and outputs are described using the same set of wildcards that will be substituted upon execution. The Snakemake engine builds the rules into computational directed acyclic graphs (DAGs), whose vertices are the instantiated rules and a directed edge (a,b) represents the fact that the input of the rule b requires the output of the rule a [14]. The rule instances are subsequently executed in the order given by a topological sort of the DAG.

SnakeLines [4] is a collection of configurable computational pipelines that extends the Snakemake workflow engine to allow the user (possibly a laboratory scientist without the ability to script its own pipelines) to define a sequence of steps of the analysis in a comprehensible YAML configuration file. SnakeLines then loads the rules required for each step, lets Snakemake build and execute the computational DAG and in the end, generates the summarizing reports and archives them together with the output log, configuration file and other necessary metadata. The archivation step ensures full reproducibility and is especially useful when the number of different analyses on various datasets grows, as it becomes much more difficult to manually keep track of the data paths and executed commands.

Each of the SnakeLines rules describes a particular atomic operation of the analysis, e.g., deduplication, mapping, etc., following the convention that each rule should use a single third-party tool (after which it is called, see Fig. 2) or have a clear, easily describable purpose. As the environmental dependencies of the tools differ and often are in conflict, each rule operates in its own Conda environment. Conda (`https://conda.io`) is a package management tool for Python supported by Snakemake to take care of installation of the desired package version and its execution in an isolated environment, which greatly simplifies its deployment.

The Snakemake engine natively supports parallel execution on any number of provided cores. Each rule can specify the number of cores required for its execution according to which Snakemake engine schedules the run. As a consequence, SnakeLines can be quickly deployed on a wide range of computational infrastructures from regular personal computers to different high-performance computers (HPC).

To deploy our pipeline, one must download the SnakeLines framework (preferably using Conda, alternatively by cloning the Git repository) and the YAML configuration file that defines the analysis. After specifying the set of input files to be analyzed using a regular expression, the Snakemake engine is initiated with the number of provided cores.

3.2 SARS-CoV-2 Pipeline: Tools and Settings

As SARS-CoV-2 genomes typically contain mostly short mutations, we have decided to pursue a reference-based approach. For the analysis of the Illumina COVIDSeq samples, we devised the following pipeline. The reads preprocessing starts with the trimming step using the Cutadapt tool [19], which is configured to first remove the PCR primers from both sides of the read, then cut off those

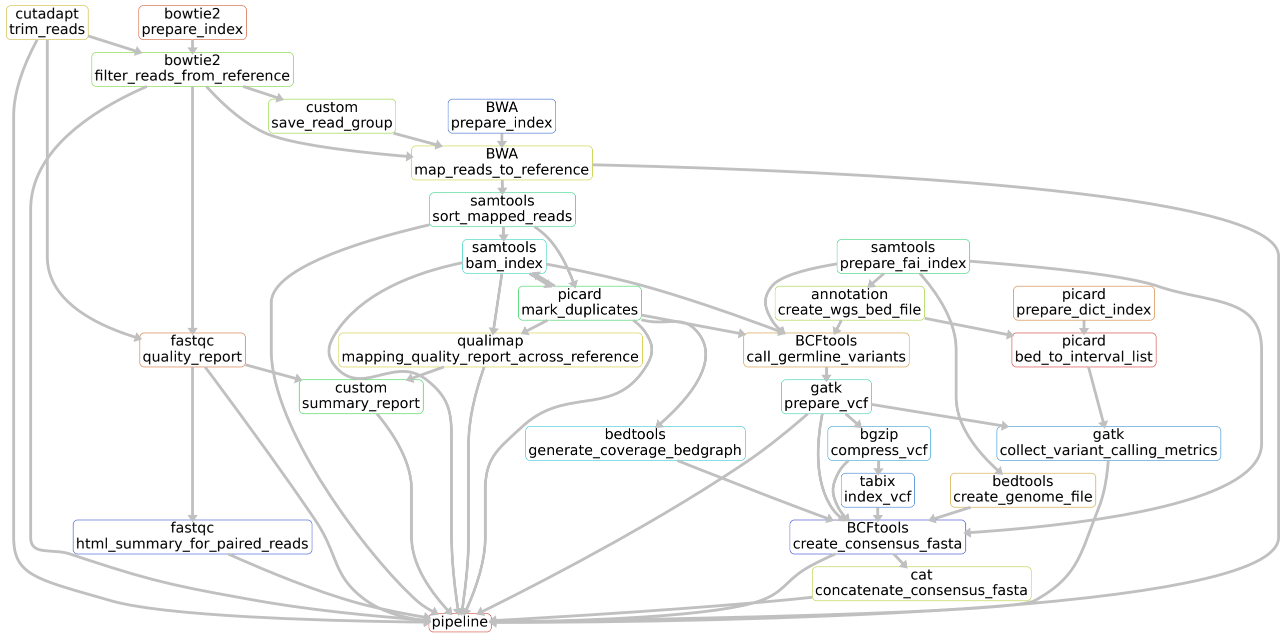

Figure 2: The computational DAG of the SARS-CoV-2 variant calling pipeline implemented in the SnakeLines engine. Each vertex corresponds to a Snakemake rule, while a directed edge represents input/output dependence. The 'pipeline' rule serves to unite the files that should be generated.

bases from both ends of the reads whose quality is lower than 20. The reads that end up with fewer than 20 bases after the trimming are deemed too short and will be removed. The set of primers together with the cutting thresholds is fully configurable.

Subsequently, the reads are subjected to a decontamination process, in which the fragments of human RNA are eliminated. This is done by aligning the reads to the human genome using the Bowtie2 [15] tool and removing those that were successfully mapped. Afterwards, the reads that contain the same fragments, presumably due to the PCR amplification, are removed in the deduplication step using the FastUniq [35] tool. The reads preprocessing is finalized by the generation of summary reports assessing the reads characteristics using the FastQC [3] toolset.

In the mapping step, we employed the BWA [17] tool to map the reads to the SARS-CoV-2 reference genome and SAMtools [18] to sort and index the generated SAM/BAM files. Reads that originate from the same DNA fragment and were duplicated by PCR are removed by the Picard [12] tool. Again, the final step is the quality statistics summarization with the use of the Qualimap 2 [21] tool. Variant calling is performed by the BCFtools [18] call command with the default parameters, using only reads with minimum mapping quality of 13, but without a restriction on the base quality. The quality of the variant call set is then evaluated by the GATK [24] toolset. BCFtools is also used to construct the consensus sequence, in which the 'N' base is placed wherever the coverage is less than 3. Figure 2 displays our pipeline in the form of a DAG of Snakemake jobs.

3.3 Comparison to Selected SARS-CoV-2 Pipelines

As mentioned earlier, several computational pipelines have been developed or adjusted to process SARS-CoV-2 samples. The ARTIC pipeline [20] specializes in Oxford Nanopore Technologies sequencing and offers two workflows: Medaka [30] workflow and Nanopolish [29], where the latter is aided by also using raw sequencing signals instead of only the base called reads. ARTIC covers all the steps from base calling the raw sequencing signals to variant calling and subsequent consensus building. Similarly to our SnakeLines pipeline, ARTIC utilizes Conda for its deployment.

The RECoVERY (REconstruction of COronaVirus gEnomes & Rapid analYsis) [6] pipeline operates on raw reads, regardless of the sequencing platform being used. Apart from the variant identification, RECoVERY is also able to annotate the open reading frames (ORFs) of the sequenced sample. As opposed to the command line usage of SnakeLines or ARTIC, the RECoVERY pipeline is built into the ARIES (Advanced Research Infrastructure for Experimentation in genomicS) web portal of the Galaxy platform (`https://usegalaxy.org/`), which offers a user-friendly interface with the possibility to run the analyses on the server side.

Of the mentioned pipelines, the most technologically similar to ours is the V-pipe [25], which also utilizes Conda for its deployment and the Snakemake engine for execution of its components. V-pipe accepts both single-end and paired-end Illumina NGS reads. To align the reads, V-pipe proposes a novel method called 'ngshmmalign', which is based on the profile hidden Markov models

[25] and uses LoFreq [34], ShoRAH [36] variant callers to detect single nucleotide variants.

4 Pipeline Design Issues

Modern DNA sequencing technologies opened up new possibilities for systematic characterization of viruses circulating in the population. However, the analysed data are affected by specific biases that have to be accounted for to rule out incorrect conclusions caused by laboratory artefacts. Thorough bioinformatics processing and quality control of sequenced data are therefore crucial to obtain reliable genomic sequences. Upon examination of the results that were produced by an early version of our pipeline, we needed to iteratively revisit the development to account for several identified issues. Our solutions for these problems contributed to the current version of the pipeline presented in section 3.

4.1 Primer Binding Sites

The Illumina COVIDSeq Test is designed to amplify SARS-CoV-2 virus-specific sequences with 98 amplicons, designed to cover a nearly entire viral genome. However, these primers have proven to be one of the main problems of such an approach. They remain in the reads while covering the true variability of the virus. In Figure 3 we present an example of a short genomic region containing a mutation typical for a given subtype of SARS-CoV-2 virus. Reads whose beginning overlaps the position with the mutation do not contain the mutated base because the primers were sequenced. In contrast, we see the mutation in the reads that have the position in middle or at the end (these were sequenced from an overlapping amplicon with different primer pairs). It may happen that the number of reads with a primer is much higher than the number of reads with the mutation, and as a result the mutation is not called. This is a typical problem of protocols aimed at sequencing the whole genome by amplifying sections using a pool of primers. We solved this problem by replacing the originally employed Trimmomatic tool [2] by the Cutadapt tool [19] to which we provided a list of ARTIC primers that were then removed from the reads.

4.2 Genome Coverage

Amplification of the SARS-CoV-2 genome by a large number of PCR reactions operating in two pools can lead to the formation of primer dimers. This may in turn result in sections of the genome with low coverage (Fig. 4). One solution to this problem was proposed by Itokawa et al. [13]. They introduced 12 alternative primers in the ARTIC primer set to replace primers that were predicted to be involved in 14 primer interactions which improved the results. Another solution, which we have used, is to mark the non-sequenced sites by the ‘N’ symbol in the consensus sequence, which means that this site had low coverage. The proportion of genome bases masked as ‘N’ in reported samples varied between 0.01%–9.69% with the median of 0.25%.

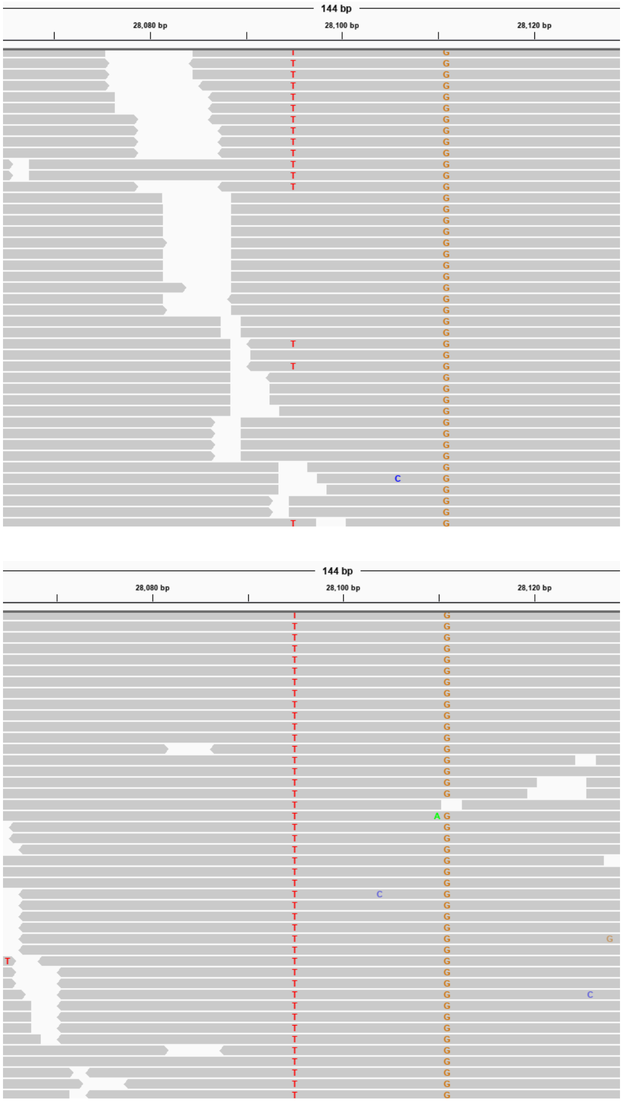

Figure 3: Example of a mutation (A28095T) that was undetected while using Trimmomatic [2] (top) and the same mutation after employing Cutadapt [19] (bottom). Nearby mutation does not overlap a primer and was detected in both cases. The visualization was performed with IGV software [26].

4.3 Problematic Regions for Mapping and Variant Calling

The third issue represents problematic regions for mapping and variant calling. Most mappers and variant callers had a huge problem to correctly determine these problematic sites. For example, region 11288–11296 (Fig. 5, top)

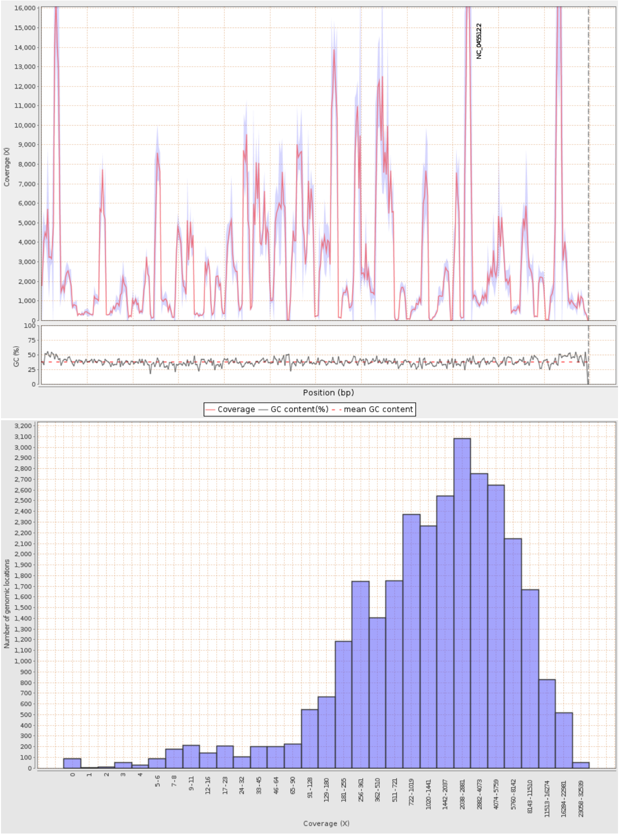

Figure 4: Example of a genome coverage across a reference of SARS-CoV-2 sample (top) and an example of genome coverage histogram of SARS-CoV-2 sample (bottom). The plots were generated by Qualimap 2 [21].

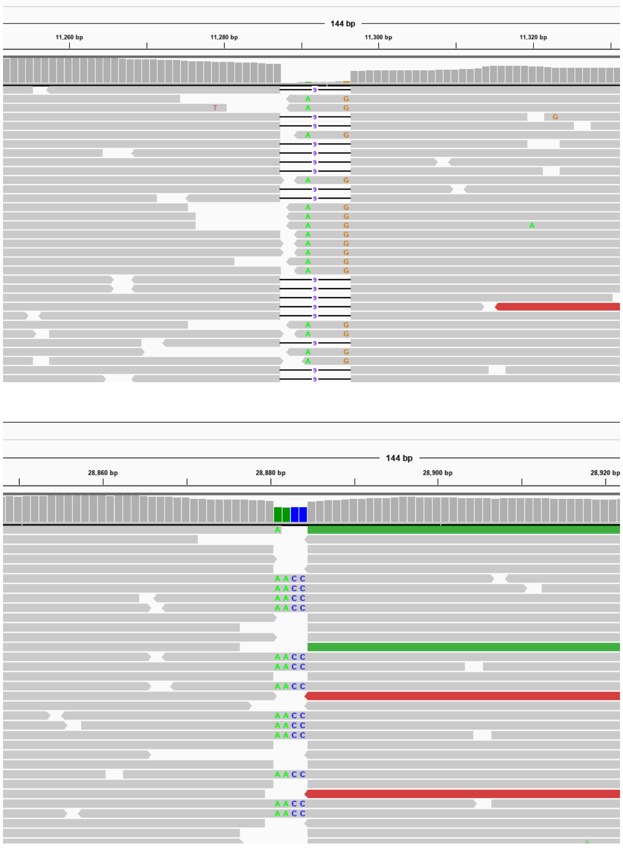

Figure 5: Visualization of the problematic regions 11288–11296 (top) and 28881–28884 (bottom) performed with IGV software [26].

	Pos. 11288-11296	Pos. 28881-28884
Freebayes	✕	✓
GATK	✓	✕
BCFtools	✓	✓

Table 1: Comparison of called variants in the problematic regions with the use of different variant callers. Positions marked with ✓ were called correctly, while positions marked with ✕ were called incorrectly.

was tricky as it contains quite a long deletion (9 bp) in the B.1.1.7 clade with four T's inside the deleted area followed by three T's and similar bases as in this deleted area. In another example, the region 28881–28884 was also problematic, as it had four mutated bases in a row in many B.1.1.7 samples. Some mappers soft-clipped the start or the end of the reads rather than mapped these mutated bases to this reference region (Fig. 5, bottom), so the variant callers also had a problem detecting these mutations. In the third example, which we see in Fig. 6, Bowtie2 mapper had a huge issue with mapping reads around the low coverage area in comparison to BWA. We resolved all the problems mentioned above with the appropriate selection of mappers and variant callers. The performance of individual variant callers at the problematic regions is outlined in Table 1.

5 Conclusions and Future Work

In this paper we present a computational pipeline for the analysis of the SARS-CoV-2 sequencing data obtained by Illumina NGS. The pipeline is highly flexible due to the full interchangeability of its components and parametrization of individual tools through a simple YAML configuration file. All the required tools are installed automatically in an isolated virtual environment, which enables rapid setup on fresh systems and reproducibility across different Unix-based systems.

We have successfully deployed our pipeline on two computational clusters within Comenius University in Bratislava (CU), namely, one at the CU Science Park and one at the Slovak Centre of Scientific and Technical Information. To this day (from mid-February to late July of 2021), we have successfully sequenced and analyzed 3055 samples with sufficient coverage (at least 3 bp across more than 90% of the genome) which were then released in dedicated public repositories - European Nucleotide Archive (ENA) and GISAID. Our pipeline continues to be rou-

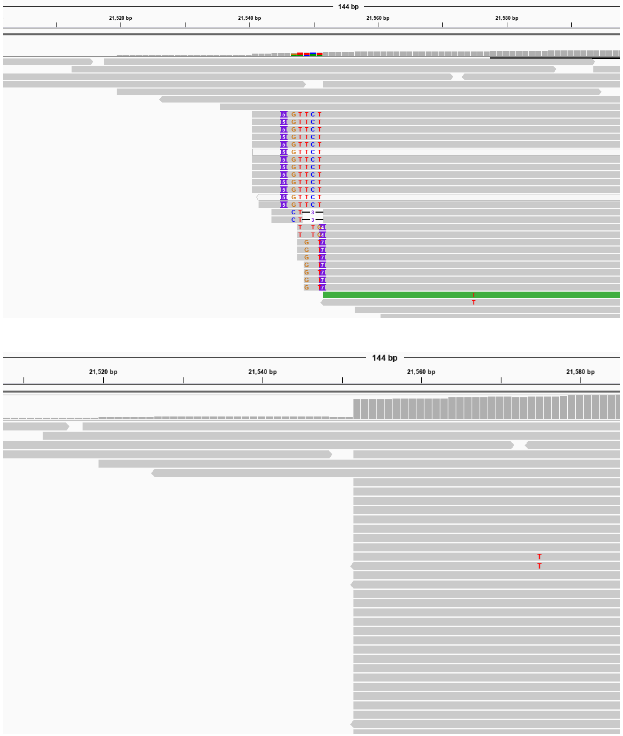

Figure 6: Visualization of the problematic region with low sequencing coverage (21546–21552) mapped with Bowtie2 (top) and BWA (bottom) performed with IGV software [26].

tinely used for national genomic surveillance of SARS-CoV-2 in Slovakia orchestrated by the Public Health Authority of Slovak Republic.

We also cover our experience with the design and tuning of the pipeline development. Each of the design choices discussed in section 4 originated in response to the particular problems that arose during the testing. For future work of this kind, we recommend to include a systematic evaluation of individual tools and thoroughly validate pipelines in isolated environments before their deployment.

The pipeline we presented was originally designed exclusively for the Illumina paired-end reads. However, SnakeLines was recently extended to support the single-end reads from ONT sequencers, which paved the way for us to design an ONT version of our pipeline, which is currently in development.

Data Availability

The SnakeLines framework is freely available for non-commercial users at `https://github.com/jbudis/snakelines` along with Anaconda repository `https://anaconda.org/bioconda/snakelines`.
Instructions for installation, running, and extending the framework, are accessible from the online documentation `https://snakelines.readthedocs.io/`.
The configuration and an example of the presented SARS-CoV-2 Variant Detection pipeline is described in `https://snakelines.readthedocs.io/en/latest/pipelines/covidseq.html`.
Real-world sequencing data for the analysis can be downloaded from the ENA portal (`https://www.ebi.ac.uk/ena/browser/view/PRJEB43444`).

Acknowledgment

This publication has been published with the support of the Operational Program Integrated Infrastructure within projects: Pangenomics for personalized clinical management of infected persons based on identified viral genome and human exome (Code ITMS:313011ATL7, 80%); and “Horizontal ICT support and central infrastructure for research and development institutions” (Code ITMS: 313011F988, 10%), both co-financed by the European Regional Development Fund. It was also supported by Comenius University Grant UK/405/2021 (5%) and a grant from VEGA 1/0458/18 (5%).

References

[1] Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A Gurevich, Mikhail Dvorkin, Alexander S Kulikov, Valery M Lesin, Sergey I Nikolenko, Son Pham, Andrey D Prjibelski, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5):455–477, 2012.

[2] Anthony M Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014.

[3] Joseph Brown, Meg Pirrung, and Lee Ann McCue. FQC Dashboard: integrates FastQC results into a web-based, interactive, and extensible FASTQ quality control tool. *Bioinformatics*, 33(19):3137–3139, 2017.

[4] Jaroslav Budis, Werner Krampl, Marcel Kucharik, Rastislav Hekel, Adrian Goga, Michal Lichvar, David Smolak, Miroslav Bohmer, Andrej Balaz, Frantisek Duris, Juraj Gazdarica, Katarina Soltys, Jan Turna, Jan Radvanszky, and Tomas Szemes. SnakeLines: integrated set of computational pipelines for sequencing reads. arXiv 2106.13649, 2021.

[5] Simone Ciccolella, Luca Denti, Paola Bonizzoni, Gianluca Della Vedova, Yuri Pirola, and Marco Previtali. MALVIRUS: an integrated web application for viral variant calling. bioRxiv 2020.05.05.076992, 2020.

[6] Luca De Sabato, Gabriele Vaccari, Arnold Knijn, Giovanni Ianiro, Ilaria Di Bartolo, and Stefano Morabito. SARS-CoV-2 RECoVERY: a multi-platform open-source bioinformatic pipeline for the automatic construction and analysis of SARS-CoV-2 genomes from NGS sequencing data. bioRxiv 2021.01.16.425365, 2021.

[7] Philip Ewels, Måns Magnusson, Sverker Lundin, and Max Käller. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, 32(19):3047–3048, 2016.

[8] Erik Garrison and Gabor Marth. Haplotype-based variant detection from short-read sequencing. arXiv 1207.3907, 2012.

[9] Alexey Gurevich, Vladislav Saveliev, Nikolay Vyahhi, and Glenn Tesler. QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, 2013.

[10] Illumina. Understanding Illumina Quality Scores. *Technical Note: Informatics*, 23, 2014.

[11] Illumina. An Introduction to Next-generation Sequencing Technology. `https://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf`, 2015.

[12] Broad Institute. Picard toolkit. `http://broadinstitute.github.io/picard/`, 2019.

[13] Kentaro Itokawa, Tsuyoshi Sekizuka, Masanori Hashino, Rina Tanaka, and Makoto Kuroda. Disentangling primer interactions improves SARS-CoV-2 genome sequencing by multiplex tiling PCR. *PLOS One*, 15(9):e0239403, 2020.

[14] Johannes Köster and Sven Rahmann. Snakemake — a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19):2520–2522, 2012.

[15] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359, 2012.

[16] Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, and Tak-Wah Lam. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, 31(10):1674–1676, 2015.

[17] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv 1303.3997, 2013.

[18] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[19] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1):10–12, 2011.

[20] ARTIC Network. The ARTIC field bioinformatics pipeline. `https://github.com/artic-network/fieldbioinformatics`.

[21] Konstantin Okonechnikov, Ana Conesa, and Fernando García-Alcalde. Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data. *Bioinformatics*, 32(2):292–294, 2016.

[22] Áine O'Toole, Emily Scher, Verity Hill, and Andrew Rambaut. Pangolin - Phylogenetic Assignment of Named Global Outbreak LINeages. `https://github.com/cov-lineages/pangolin`, 2020.

[23] Nicole Pedro, Cláudio N Silva, Ana C Magalhães, Bruno Cavadas, Ana M Rocha, Ana C Moreira, Maria Salomé Gomes, Diogo Silva, Joana Sobrinho-Simões, Angélica Ramos, et al. Dynamics of a Dual SARS-CoV-2 Lineage Co-infection on a Prolonged Viral Shedding COVID-19 Case: Insights into Clinical Severity and Disease Duration. *Microorganisms*, 9(2):300, 2021.

[24] Ryan Poplin, Valentin Ruano-Rubio, Mark A DePristo, Tim J Fennell, Mauricio O Carneiro, Geraldine A Van der Auwera, David E Kling, Laura D Gauthier, Ami Levy-Moonshine, David Roazen, et al. Scaling accurate genetic variant discovery to tens of thousands of samples. bioRxiv 201178, 2018.

[25] Susana Posada-Céspedes, David Seifert, Ivan Topolsky, Kim Philipp Jablonski, Karin J Metzner, and Niko Beerenwinkel. V-pipe: a computational pipeline for assessing viral genetic diversity from high-throughput data. *Bioinformatics*, 2021.

[26] James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. Integrative genomics viewer. *Nature Biotechnology*, 29(1):24–26, 2011.

[27] AE Samoilov, VV Kaptelova, AY Bukharina, OY Shipulina, EV Korneenko, AV Lukyanov, AA Grishaeva, AA Ploskireva, Anna S Speranskaya, and VG Akimkin. Change of dominant strain during dual SARS-CoV-2 infection. medRxiv 2020.11.29.20238402, 2020.

[28] Yuelong Shu and John McCauley. GISAID: Global initiative on sharing all influenza data–from vision to reality. *Eurosurveillance*, 22(13):30494, 2017.

[29] Jared Simpson. Nanopolish: Signal-level algorithms for minION data. `https://github.com/jts/nanopolish`, 2018.

[30] Oxford Nanopore Technologies. Medaka. `https://github.com/nanoporetech/medaka`, 2018.

[31] Changtai Wang, Zhongping Liu, Zixiang Chen, Xin Huang, Mengyuan Xu, Tengfei He, and Zhenhua Zhang. The establishment of reference sequence for SARS-CoV-2 and variation analysis. *Journal of Medical Virology*, 92(6):667–674, 2020.

[32] Ryan R Wick, Louise M Judd, Claire L Gorrie, and Kathryn E Holt. Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. *PLOS Computational Biology*, 13(6):e1005595, 2017.

[33] Ryan R Wick, Mark B Schultz, Justin Zobel, and Kathryn E Holt. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*, 31(20):3350–3352, 2015.

[34] Andreas Wilm, Pauline Poh Kim Aw, Denis Bertrand, Grace Hui Ting Yeo, Swee Hoe Ong, Chang Hua Wong, Chiea Chuen Khor, Rosemary Petric, Martin Lloyd Hibberd, and Niranjan Nagarajan. LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic Acids Research*, 40(22):11189–11201, 2012.

[35] Haibin Xu, Xiang Luo, Jun Qian, Xiaohui Pang, Jingyuan Song, Guangrui Qian, Jinhui Chen, and Shilin Chen. FastUniq: a fast de novo duplicates removal tool for paired short reads. *PLOS One*, 7(12):e52249, 2012.

[36] Osvaldo Zagordi, Arnab Bhattacharya, Nicholas Eriksson, and Niko Beerenwinkel. ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC Bioinformatics*, 12(1):1–5, 2011.