

Facilitating Rapid Screening of Metaviral Data with the VERA Pipeline

Chudinov I.K.^{*1,2}, Speranskaya A.S.³

¹*Moscow Institute of Physics and Technology (National Research University), Moscow, Russia*

²*Scientific Research Institute for Systems Biology and Medicine, Federal Service on Consumer Rights Protection and Human Well-Being Surveillance, Moscow, Russia*

³*Vavilov Institute of General Genetics Russian Academy of Science, Moscow, Russia*

Abstract. The study of metaviral data has become a crucial area of research in recent years, leading to a significant increase in metagenomic and metaviral datasets. This growth has resulted in advancements in data collection and processing techniques, focusing on analyzing the complex viral diversity found in natural samples. Recent developments in metaviral research have emphasized the integration of high-throughput sequencing technologies with advanced bioinformatics tools, enhancing the sensitivity and accuracy of virus identification. This study introduces a novel pipeline designed for rigorous double filtering of assembled contigs, maximizing the detection of true viral sequences while minimizing the inclusion of non-viral artifacts. By improving virus detection through dual taxonomic filtering and leveraging modern workflow management systems, the new pipeline not only addresses gaps in existing methods but also offers a fresh approach to metaviral research. The VERA pipeline was implemented using the Nextflow workflow system, and its validation was performed with control samples, showing consistent results that align with manual validation and commercial platforms. This work highlights the importance of innovative approaches in capturing the full breadth and complexity of viral diversity, even amidst recent technological advancements.

Key words: NGS, metaviral, nextflow, pipelines

INTRODUCTION

Over the past four decades, there have been several viral outbreaks and global pandemics, including but not limited to Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV), Influenza H1N1 (2009-2010), Middle East Respiratory Syndrome Coronavirus (MERS-CoV) and Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2, 2020-2023) [1]. The collection and analysis of metaviral data has emerged as a crucial step toward understanding the complex diversity of viruses circulating in natural reservoirs. The number of studies has revealed that the vast majority of viral pandemics are zoonotic in origin [2].

With the advancements in next-generation sequencing (NGS), researchers have obtained powerful tools for the low cost and relatively fast detection of known zoonotic and novel viruses, for example, [3, 4]. In addition to the efforts of individual research groups, global initiatives such as the Global Virome Project [5] have led to a substantial increase in metaviral datasets, emphasizing the need for the development of reliable and precise pipelines for assembling and detecting viruses from complex datasets.

The bioinformatics community has increasingly adopted dataflow languages and workflows to simplify and unify approaches for common and complex analyses. Tools such as Snakemake

*ikchudinov@gmail.com

[6] or Nextflow [7] have become integral to the development of bioinformatics pipelines for broad purposes, including metaviral data processing. These platforms emphasize the principles of reproducibility and scalability, thus enabling the seamless integration of a variety of bioinformatics tools into workflows. The widespread adoption of community-curated pipelines has significantly advanced the field, as evidenced by the utility of dataflow languages and their ability to enhance efficiency and reliability in the processing of large-scale metagenomic and metaviral datasets [8].

In the context of metagenomic and metaviral analysis, a range of software tools has been developed for the purposes of identifying and characterizing viral communities within complex samples. These tools include the *nf-core/viralrecon*, which is specifically designed for intra-host/low-frequency variant calling for viral samples [9], and the VIBRANT, which facilitates the automated recovery and annotation of bacterial and archaeal viruses [10]. The ViromeScan project also offers an approach for metagenomic viral community profiling, while the GenomeDetective pipeline is a commercial solution that incorporates metaviral and metagenomic datasets for processing and analysis [11, 12]. The machine learning-based tools like VirFinder and DeepVirFinder leverage nucleotide composition features (e.g., k-mer frequencies) to predict viral sequences in metagenomic data [13].

Protein-based approaches identify short protein sequences (motives), which are often more conserved than nucleotide sequences, which makes them particularly useful for detecting divergent or novel viruses. These methods typically involve homology searches against curated databases of viral proteins, such as the NCBI Viral RefSeq database or custom databases tailored to specific viral families [14, 15]. A protein-based approach for identifying viruses in the absence of a common marker was previously successfully implemented by Paez-Espino *et al.*, who emphasized the importance of refined methodologies for improving the accuracy of virus detection in metagenomic samples [16]. VirSorter2, based on the principles of machine learning established by tools like VirFinder and DeepVirFinder, extends these approaches by integrating protein-based features (e.g., hallmark genes and protein domains) with nucleotide and structural characteristics to improve the accuracy and scope of virus detection. It employs a combination of machine learning models and curated databases (protein-based features combined with nucleotide-level information). It classifies sequences into different categories based on their likelihood of being viral [17]. It is designed to integrate into existing workflows, making it a useful tool for metaviral research, as shown by Bukin *et al.* [18].

In the context of the rapidly expanding field of metaviral research, this study presents the successful creation and validation of a novel metaviral processing pipeline, VERA. The novel pipeline builds on our previous research and addresses longstanding challenges in metaviral data analysis [4, 19, 20, 21]. This pipeline incorporates advances in dataflow languages and rigorous double taxonomy filtration, maximizing the detection of true viral sequences and minimizing the inclusion of non-viral artifacts. The novel pipeline not only bridges the gap between existing methods but also proposes new approaches for metaviral studies.

1. METHODS

1.1. Pipeline implementation

The computational pipeline VERA is implemented on Nextflow [7], utilizing a combination of self-built and off-the-shelf *nf-core* modules [8]. Modules from *nf-core* include bowtie2 [22], cat/fastq, custom/dumpsoftware [8, 23], fastp [24], fastqc [25], gunzip [26], kraken2 [27], multiqc [28], samtools [29], seqkit [30, 31] and spades [32]. Self-built modules represent two groups – *mmseqs/easytaxonomy* and a group of processes defined to launch Python-based [33] parsers.

The pipeline is freely available at <https://github.com/risbm-lmi/wf-vera>.

1.2. Local modules

1.2.1. Taxonomy

Local module `mmseqs/easytaxonomy` is inherited from `nf-core mmseqs/easy-search` [8, 23], including container image and resources label. The process takes two channels as input, namely path to the FASTA-file to be classified and path to the databases of interest. The process outputs five channels, four of which correspond to the standard `mmseqs easytaxonomy` workflow [34, 35] including least common ancestor information, taxonomy report in `kraken` style, top hit report and top hit alignment information. A fifth channel contains a file with versions of the packages used in the process according to the codestyle of `nf-core`. As in the `mmseqs/easy-search nf-core` module, the process derives the target database name from path and launches `mmseqs easytaxonomy` workflow.

Additional arguments are provided apart from `nf-core mmseqs/easy-search`, namely `split-memory-limit` to prevent interference during parallel computations of samples within the workflow.

Both for protein and nucleotide processes the least common ancestor ranks are set as "superkingdom,order,family,genus" for convenient results processing and successful automated parsing. The results of the nucleotide stage are additionally preformatted to output information about query and subject hits.

1.2.2. Parsers

All the processes withing the `parsers` use `BioContainers` [36, 37] with at least `python` (v3.9.6) and `pandas` (v1.3.0) package [33, 38]. Local module `mmseqs2table_protein` takes a single input channel with the least common ancestor table from `mmseqs/easytaxonomy`, launches the python script for taxonomy inference and outputs three channels containing a TXT file with the identified as viruses FASTA headers, a table with the taxonomy information and a file with software versions. Local module `mmseqs2table` takes two input channels with the `mmseqs/easytaxonomy` least common ancestor information and the table with the alignments, launches the python script for taxonomy inference and outputs two channels containing a table with the selected identified viral records and a file with the software versions. Local module `create_extended_table` takes as input one channel with nucleotide and protein taxonomy inference tables, table with contigs coverage from `samtools` process, launches a python script for tables merging and outputs three channels, namely the table with the taxonomy identification results, a list of contigs headers with the potential for extension, a tab-delimited file with contigs coverage and a file with software versions.

1.3. Resources

1.3.1. Databases and references

The following databases and references were used during the pipeline development and in the current study:

- *Pipistrellus kuhlii* genome sequence (accession GCF_014108245.1) derived from GenBank with `ncbi_datasets` [14].
- Severe acute respiratory syndrome coronavirus 2 genome sequence (accession NC_045512.2) derived from GenBank directly.
- `kraken2` [27] default databases namely `bacteria`, `viral`, `archaea`, `fungi`, `protozoa` complemented by genome sequences of *Homo sapiens*, *Mus musculus*, *Rattus norvegicus*, *Drosophila melanogaster*, *Erinaceus europaeus*, *Pipistrellus kuhlii*, *Bos taurus*, *Salmo*

salar and chloroplast sequences of *Viridiplantae* clade, database assembly 23 November 2023.

- Protein pre-clustered database UniRef90 [39] for initial taxonomy filtration obtained with `mmseqs` databases [34, 35], database obtained 03 October 2024.
- Nucleotide database RefSeq Viral manually created from RefSeq Viral [14, 15] sequences with `mmseqs createdb` [34, 35], as described in Appendix A, database created 06 September 2024.

1.3.2. SRA data

The following SRA [40] records were downloaded with `sra-tools` [41]: SRR15525307, SRR15524477.

1.3.3. Computations

All the calculations with VERA were performed on a high-performance computing (HPC) cluster with two AMD EPYC 7642 48-Core Processors, 1 Tb of RAM and an SSD disk with the `queueSize = 8` nextflow configuration.

All the calculations with `Genome Detective` were performed with cloud resources provided in the platform personal account.

1.4. Results curation

The results of VERA pipeline and the `Genome Detective` tool were curated with the review of the assembled contigs and their taxonomy identification tables using jupyter notebook [42].

To check the integrity of the selected contigs among the pipelines `mmseqs easy-cluster` [34, 35] module was used with `cluster-reassign`, `cov-mode 1` and `cluster-mode 2` options.

The clustering results were processed with `pandas` (v2.2.0) [43, 38] package and visualised with `matplotlib` (v3.8.2) [44] and `networkx` (v3.2.1) [45] packages. All figures were created with Python [33] using `numpy` (v1.26.3) [46], `pandas` (v2.2.0) [43, 38], `matplotlib` (v3.8.2) [44], and `seaborn` (v0.13.2) [47] with the following InkScape [48] and/or PDFgear [49] recompilation.

2. RESULTS

2.1. Pipeline overview

The computational pipeline was implemented using Nextflow [7] in the terms of the `nf-core` [8] style and with the use of community-curated and local modules.

The VERA computational pipeline (see Fig. 1) comprises two distinct modules: the initial processing of metaviral short reads, followed by their subsequent assembly (outlined in the upper area by the blue dashed line), and the subsequent analysis of the obtained sequences (outlined in the lower area by the magenta dotted line). The data necessary to execute the pipeline (highlighted in blue in the schematic diagram) encompasses a table with sample metadata in CSV format, along with raw short reads corresponding to the samples enumerated in the table. Additionally, one optional (`kraken2`) and two mandatory (protein UniRef and nucleotide RefSeq Viral in `mmseqs2` format) databases are required for executing the pipeline, along with pre-indexed genomic sequences of the host (an animal from which the swab was collected). The pre-indexed genomic sequences of the known intra-laboratory contaminant organisms are also optional and recommended (for example human, rat etc.). The optional data processing steps are denoted by a gear symbol in the schematic diagram (see Fig. 1). The result of the processing pipeline (marked in green on the schematic) is sequences selected on the basis of taxonomic

identification from the set of all collected contigs, as well as summary tables with a preliminary taxonomic conclusion for each of the analyzed samples.

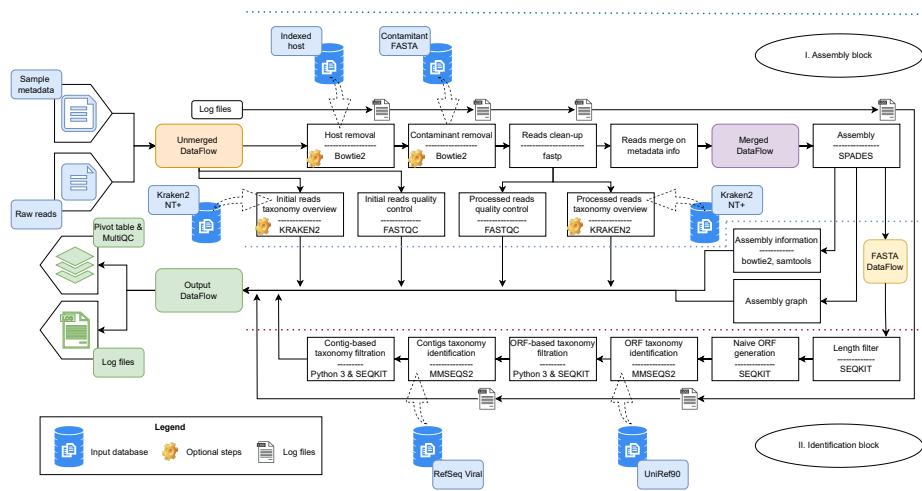


Fig. 1. Pipeline logic overview. [Full size image.](#)

At the pipeline entry point, the input data received on the samples under investigation are combined into a DataFlow type structure for sequential or parallel processing by the pipeline modules. The first processing steps are taxonomic identification of reads by the *kraken2* tool [27], quality control of reads and purification from host or in-laboratory contaminant. The next step is semi-automated read processing to filter out low quality reads and remove technical subsequences from short reads. Following the initial cleaning and preparation of the data, the taxonomic analysis and quality assessment of the reads is conducted again to enhance the clarity of the results obtained from the data preparation process. If reads from the same sample obtained from different runs, the above steps are performed for different runs independently, without combining of reads. Data analysis and cleaning are separated for reads obtained during different sequencing runs. This is necessary to build the researcher's knowledge of the correctness of animal sampling and purity of the original sample, and estimation of the possibility of lab contamination during libraries preparation experiments and sequencing, the success of the sequencing and the purity of the processed data, which is essential for expert judgment in analyzing the results. Clean reads are merged between runs according to sample metadata and fed into a new DataFlow channel, which is passed to *de novo* assembly using *spades* [32]. The contigs resulting from the *de novo* assembly of short reads are directed to the DataFlow channel for taxonomic identification, and subsequently sent for contig coverage analysis. The results of the coverage analysis, in conjunction with the *de novo* assembly graph, are transmitted to the output data channel.

The data channel processing for taxonomic identification includes pre-filtering of contigs using *seqkit* [30, 31] of short artifactual sequences which are expected not to carry significant information (100 bp and less). Then all possible amino acid sequences over six reading frames are generated. The obtained amino acid sequences are then sent to the *mmseqs/easytaxonomy* workflow based on the LCA concept [34, 35] for the first step of taxonomic identification using the UniRef database [39]. Subsequent to this, contigs for which the generated amino acid sequences can be potentially attributed to virus proteins are selected for the second step of taxonomic identification which is performed against the RefSeq Viral database. Sequences confirmed as viral during the second taxonomic identification step are selected for transmission to the output using *seqkit*. Each stage of the data processing pipeline collects the log files of

each of the programs used into a separate data channel. Thereafter, all log files are also forwarded to the output data channel.

We found through experiments (the data obtained are not presented in this article) that a two-stage taxonomic identification and filtering process is necessary because metaviral data often contains fragments of eukaryotic, archaeal, and prokaryotic genomes. If only a single filtering step against the viral database is used, it results in a large number of false positive taxonomic assignments. This complicates result handling and significantly increases analysis time. Conventionally, the utilization of `blastn` [50] with the comprehensive RefSeq database [14, 15] exerts substantial demands on available disk space. However, the employment of the pre-clustered UniRef database [39] offers a substantial reduction in the requisite disk space by a factor of ~ 3 , making the computational pipeline applicable across different platforms. The two-stage approach serves as a taxonomic filter in the initial stage and as an identification process in the subsequent stage. It is also noteworthy that utilizing `mmseqs2` can significantly reduce the analysis time when working with taxonomic databases, while maintaining the specificity of the `blastn` algorithm [35].

2.2. Results validation

To evaluate the efficacy of the VERA pipeline, we analyzed two previously published datasets [19, 51] and compared the results with those obtained using the `Genome Detective` platform [12]. This comparative analysis aimed to validate the reproducibility and potential enhancement of results relative to an established platform. The analysis focused on previously identified viral families [19, 51], including *Adenoviridae*, *Coronaviridae*, *Nodaviridae*, *Parvoviridae*, *Dicistroviridae*, *Astroviridae*, *Picornaviridae*, and *Nairoviridae*.

Viral contigs were extracted from both pipeline outputs based on taxonomic classification (see Fig. 2).

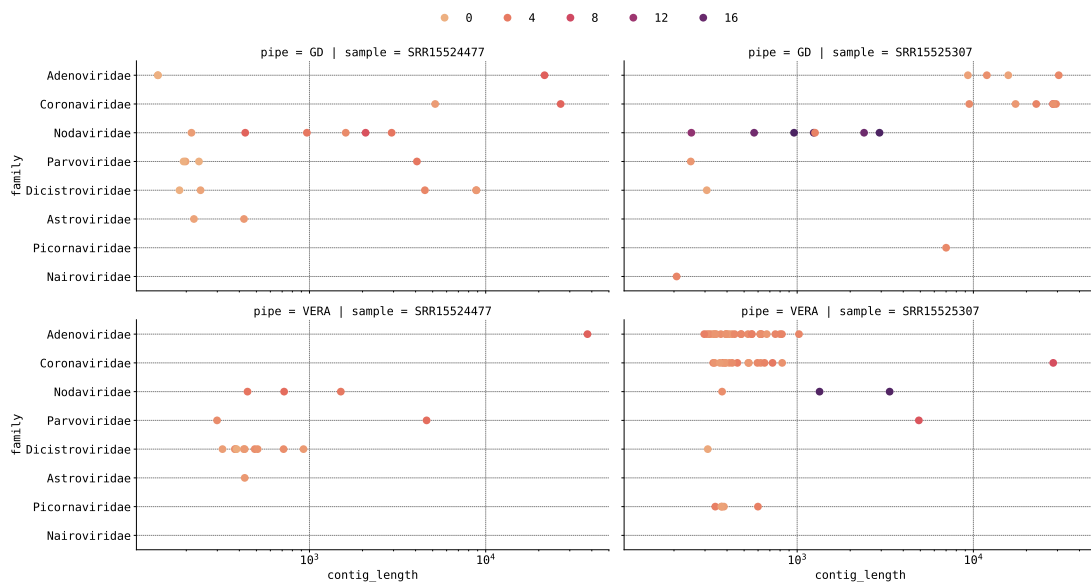


Fig. 2. Pipelines comparison. Each dot represent a contig from the corresponding viral family and length. The intensity of the dots represent $\log_2$ of a contig mean depth coverage. GD abbreviation represents results obtained with `Genome Detective`, VERA represents results obtained with the pipeline proposed. [Full size image](#).

Comparative analysis demonstrated that both pipelines successfully identified and assembled a complete *Mastadenovirus* genome in sample SRR15524477, along with *Mastadenovirus* genome fragments from *Adenoviridae*, consistent with previous findings

[19]. Additionally, both pipelines detected a semi-complete *Coronaviridae* genome in sample SRR15525307. However, while *Genome Detective* identified two *Coronaviridae* contigs (26,628 bp and 5,168 bp) in sample SRR15524477, the VERA pipeline detected none. Further investigation revealed that these contigs contained substantial proportions of unidentified nucleotides (24,895 and 4,744 respectively). Moreover, longest *Coronaviridae* contig from *Genome Detective* results is identified as a fragment of Severe acute respiratory syndrome-related coronavirus, whereas SARS-CoV-2 genome was set as a contaminant during the VERA pipeline run, as the sequencing laboratory reported routinely working with SARS-CoV-2 (inherited from personal communication), leading us to classify it as a potential contaminant [52, 53, 54, 55, 56]. Importantly, each laboratory has its own set of contaminants, and only the laboratory staff know which organisms they actively work with. Therefore, only they can provide a list of potential contaminants when analyzing sequencing data. The use of this option (removing potential contaminants) can help reduce the number of sensational claims about unexpected findings of dangerous components in biological samples, which can arise due to laboratory contamination or misinterpretation of data.

To qualitatively assess the assembly performance of both pipelines, particularly regarding contig length, we performed clustering analysis using *mmseqs easy-cluster* (see Fig. 3, Appendix B). Given that *Genome Detective* generates reference-based assemblies with gap regions denoted by N, we preprocessed these contigs by separating them into individual segments prior to clustering to prevent bias in the analysis.

Comparative analysis across viral families revealed comparable performance between the two pipelines, with some family-specific variations. For larger viral genomes, such as those in the *Adenoviridae* family in sample SRR15524477, VERA demonstrated superior performance, generating more extensive contigs that encompassed all sequences identified by *Genome Detective*. However, same viral family in sample SRR15525307 has many fragmented assemblies obtained with VERA pipeline, and *Genome Detective* platform demonstrated better performance revealing still highly fragmented but semi-complete sequence identified as *Mastadenovirus pipistrelli*. For smaller viral genomes and more fragmented assemblies, such as *Nodaviridae*, the relative performance varied between the pipelines. (see Fig. 2 and Fig. 3).

2.3. Computational performance

The pipeline performance was evaluated using two samples processed on a High Performance Computing (HPC) cluster, utilizing the *meta spades* mode with concurrent job execution limited to 8 processes.

The total execution time was 10 hours, 47 minutes, and 19 seconds. Resource utilization analysis revealed that the highest CPU consumption was attributed to three primary modules: *bowtie/align*, *mmseqs/easy-taxonomy* (protein stage), and *spades*. The memory usage profile exhibited a similar pattern, with *mmseqs/easy-taxonomy* at the protein stage demonstrating peak memory requirements (see Appendix C).

Analysis of individual module performance indicated that *spades* assembly constituted the predominant computational bottleneck, accounting for approximately 10 hours of the total execution time. This duration was primarily influenced by sample SRR15525307, requiring 9 hours and 30 minutes for assembly, while sample SRR15524477 completed in 1 hour and 46 minutes under identical *meta spades* parameters. Notably, utilizing *default spades* mode significantly reduced the total pipeline execution time to 2 hours and 21 minutes, with assembly operations completing in approximately 70 minutes.

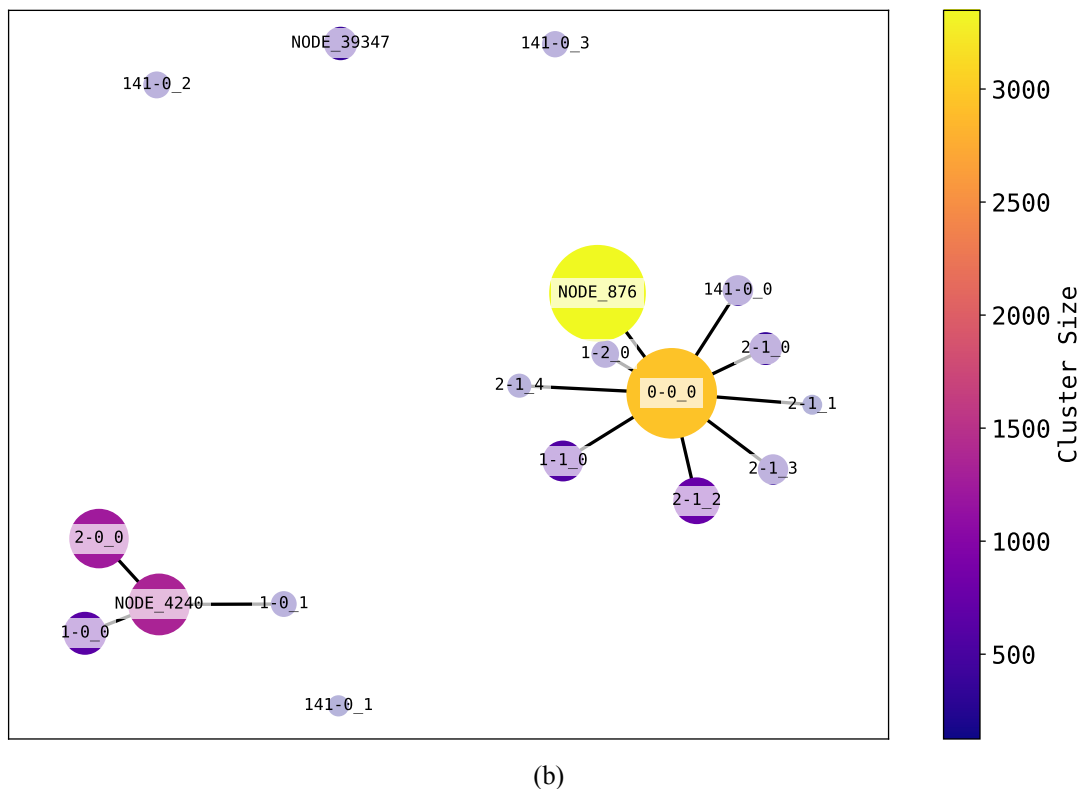
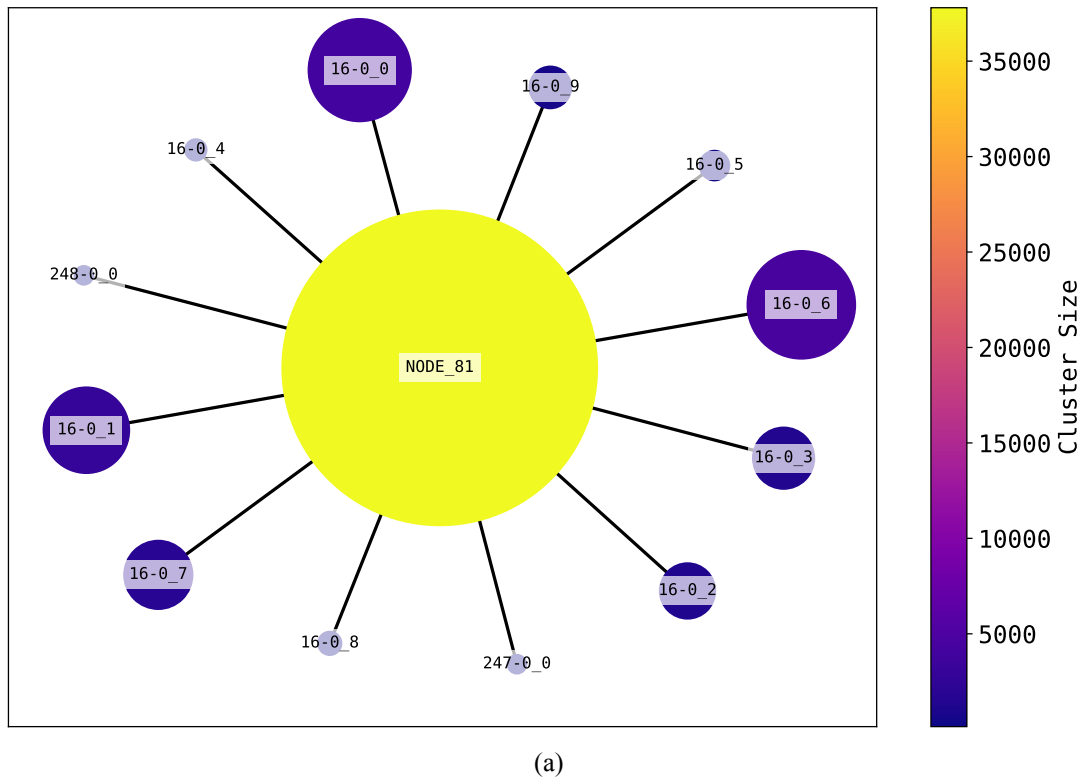


Fig. 3. Examples of contigs clustering results: a) –one distinguishable cluster of *Mastadenovirus*; b) – several separated clusters identified as *Nodaviridae*. Color and size of circles represent length of the contig, shorter contigs are clustered with similar longer ones. Names starting with NODE correspond to VERA pipeline, others to Genome Detective results. Black lines connect members of cluster leading to one longest cluster representative.

3. CONCLUSIONS

3.1. Possible improvement

This study presents a computational pipeline for the processing of metaviral data. The developed pipeline demonstrates significant scalability potential, attributed to the specific design of the chosen creation platform. The modular structure of the pipeline, facilitated by Nextflow [7] and the integration of off-the-shelf nf-core modules [8], allows for logical steps within the module—such as read cleanup from host or in-lab contaminants and *de novo* assembly—to be modified or added without necessitating a complete rebuild of the data processing pipeline. Additionally, the pipeline can be adapted to utilize exclusively long reads or hybrid assemblies.

The current methodology for preparing pipeline results includes the option for contig elongation using *cobra* [57]. Furthermore, a targeted selection of viral families can be extracted for more detailed analyses, including virus classification through *nextclade* [58] or *pangolin* [59], as well as the construction of multiple sequence alignments and phylogenetic trees for precise analyses of viral species identity.

It is important to note the sections of DataFlow highlighted in color in the schematic diagram (see Fig. 1). These sections can introduce additional input points for the pipeline, facilitating *de novo* assembly of reads pre-processed by third-party methods or enabling separate taxonomic identification of contigs obtained through such methods.

We contend that the VERA pipeline can be further refined and optimized to identify novel viruses that can be assembled from short reads but remain undetected due to the absence of closely homologous sequences in amino acid or nucleotide databases. For instance, integrating protein domain searches using *mmseqs2* [34, 35] alongside the pre-formatted Pfam database [60] could facilitate the identification of previously unknown viruses possessing specific domains that enable attachment to receptors of interest.

3.2. Metaviral research

In this study, we propose a new approach to the analysis of metaviral data. According to our benchmarking results, the pipeline is showing similar performance to the state-of-the-art tools in terms of viruses assembly and taxonomic classification. The approach for virus detection is based on the identification of contigs that contain at least one ORF that is similar to the ORFs of known viruses with later classification into the corresponding taxonomic group against a viral database. The pipeline provides results in the form of terminal contigs that can be used for further analysis and without any reference-based refinement as it is done in similar pipelines. To our knowledge, there are no pipelines that are able to detect viruses and show final results without human reviewing, so the experienced virologist efforts are still required for the final results verification. We suppose that the pipeline can be a useful tool for the virologists to quickly get the list of potential viruses in a sample and to focus on the most likely candidates for further analysis. The proposed protein-based approach allows to detect viruses that are not similar to any known viruses by the nucleotide sequence, but similar by the protein sequence.

It is important to acknowledge that the *Genome Detective* platform is a commercial product that offers a limited number of credits for trial use by new users, with additional credits required for ongoing analyses. In contrast, the pipeline proposed in this study is open source and available in accordance with nf-core best practices, allowing for installation and use by virologists without incurring additional costs. Thus, we expect that the pipeline will be most useful for initial screening of a sample and for the virologists to focus on the most likely candidates for further analysis.

APPENDICIES**A. Viral database creation**

For the construction of a custom RefSeq Viral nucleotide database, `mmseqs createdb` [34, 35] module was used. First, the online NCBI Virus resource [14, 15] was used. The `taxid:10239` was set in `Virus/Taxonomy` field. The source of the sequences was set as `RefSeq`, nucleotide completeness was set as `complete`. A TXT list of accessions (including sequence version) was retrieved from an online resource and sequences were downloaded with `ncbi_datasets` [14]. Remarkably, same sequences taxonomy selection procedures executed directly from `ncbi_datasets` result in several bacteria accessions in the dataset downloaded, thus a manual filtering of accessions with an online resource was performed.

Current NCBI taxonomy was retrieved directly with `wget` from an FTP `ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/taxdump.tar.gz`.

Next, the `seqkit` tool was used to create a TXT `accession-taxid` file. Namely, all the version numbers and descriptions were excluded from sequence headings resulting in a file of sequences' IDs only. Sequences' IDs were subjected to a chain of `edirect` [61] tools, namely `epost`, `esummary`, `xtract`, resulting in `accession-taxid` file.

The initial database was created with the `mmseqs createdb` workflow, utilizing only a FASTA file with viral sequences. Next, `createtaxdb` was used to supplement the database with the corresponding taxonomy information. The database utility was tested using `mmseqs easy-taxonomy` with different viral sequences of known taxonomy.

B. Cluster evaluation

For the *Astroviridae* family in sample SRR15524477, both pipelines — the novel VERA and the reference Genome Detective — produced similar results. Two short contigs were identified by the Genome Detective and one short contig by the VERA pipeline. One of two contigs identified by the Genome Detective was found to be a fragment of the only contig from the VERA pipeline. (see Fig. 4).

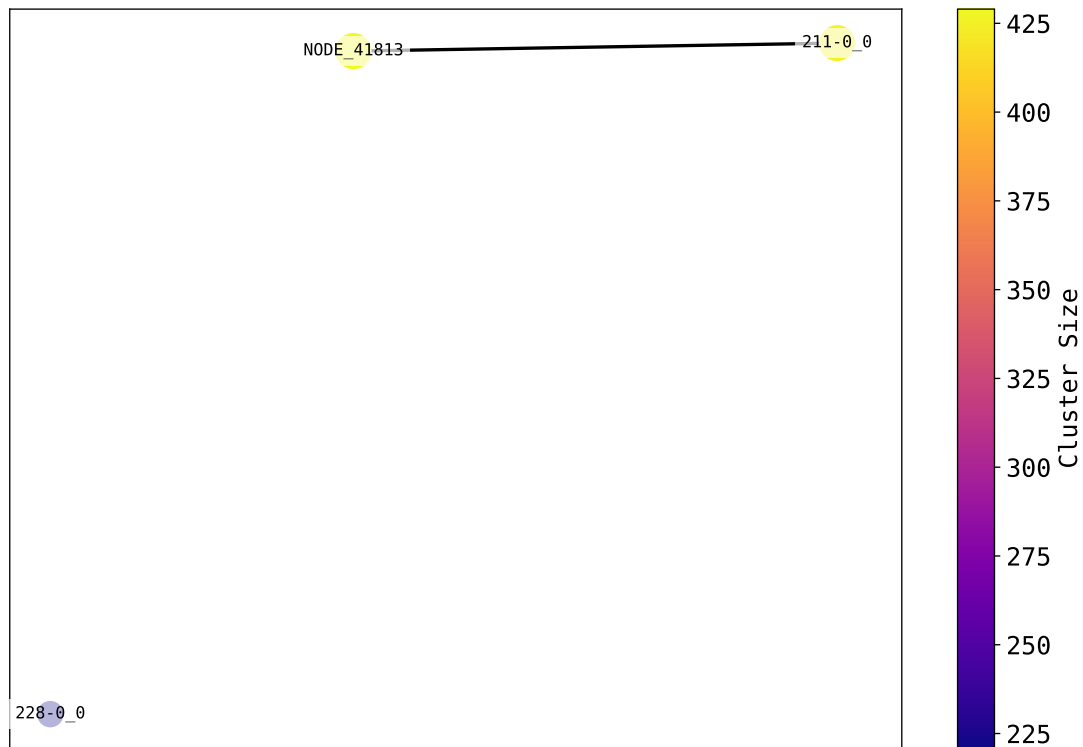


Fig. 4. Clusters evaluation for sample SRR15524477 and *Astroviridae* family.

For *Dicistroviridae* (in the sample SRR15524477), *Adenoviridae* (in the sample SRR15525307) and *Picornaviridae* (in the sample SRR15525307) both pipelines yield similar results with number of dissociated contigs (see Fig. 5, Fig. 6 and Fig. 7). The longest contigs are less than 1000 base pairs, indicating that significantly extended contigs were not obtained. Additional sequencing depth is required for the assembly of longer contigs.

The Genome Detective contig 106-0 (from sample SRR15524477) is a candidate for further PCR amplification and sequencing, as its fragments cluster with themselves and with the contig from the VERA pipeline. (see Fig. 8).

For *Parvoviridae* family (in the sample SRR15524477), VERA pipeline revealed longer contig, which include the longest contig from Genome Detective results. Other detected as *Parvoviridae* contigs are comparably short and might represent fragments of the genome that were not assembled (see Fig. 9).

Some contigs, specifically those from the *Dicistroviridae* family (in the sample SRR15525307) and the *Parvoviridae* family (in the sample SRR15523955), were not found to be similar between the pipelines or were too short for evaluation. (see Fig. 10 and Fig. 11).

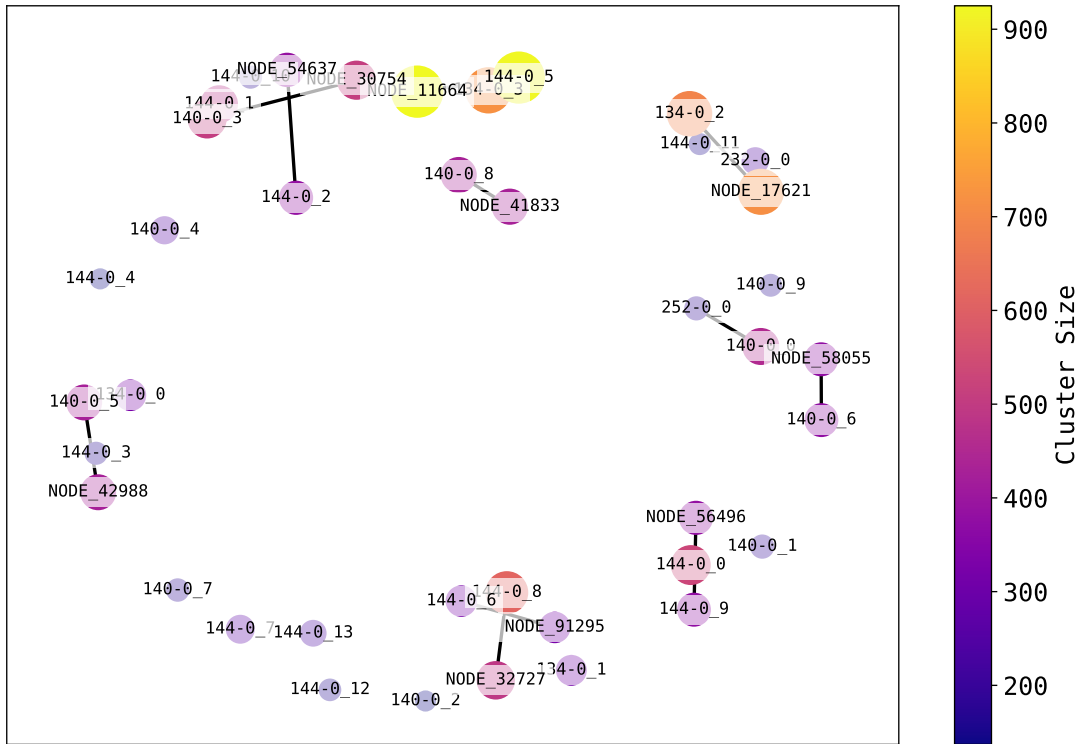


Fig. 5. Clusters evaluation for sample SRR15524477 and *Dicistroviridae* family.

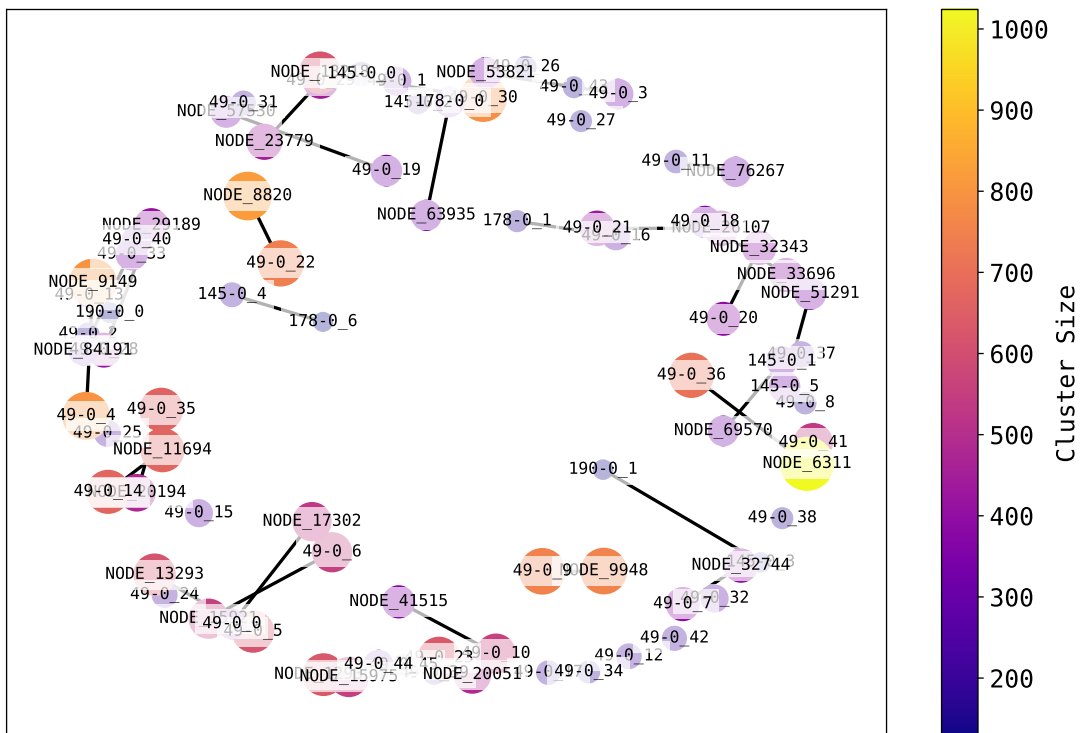


Fig. 6. Clusters evaluation for sample SRR15525307 and *Adenoviridae* family.

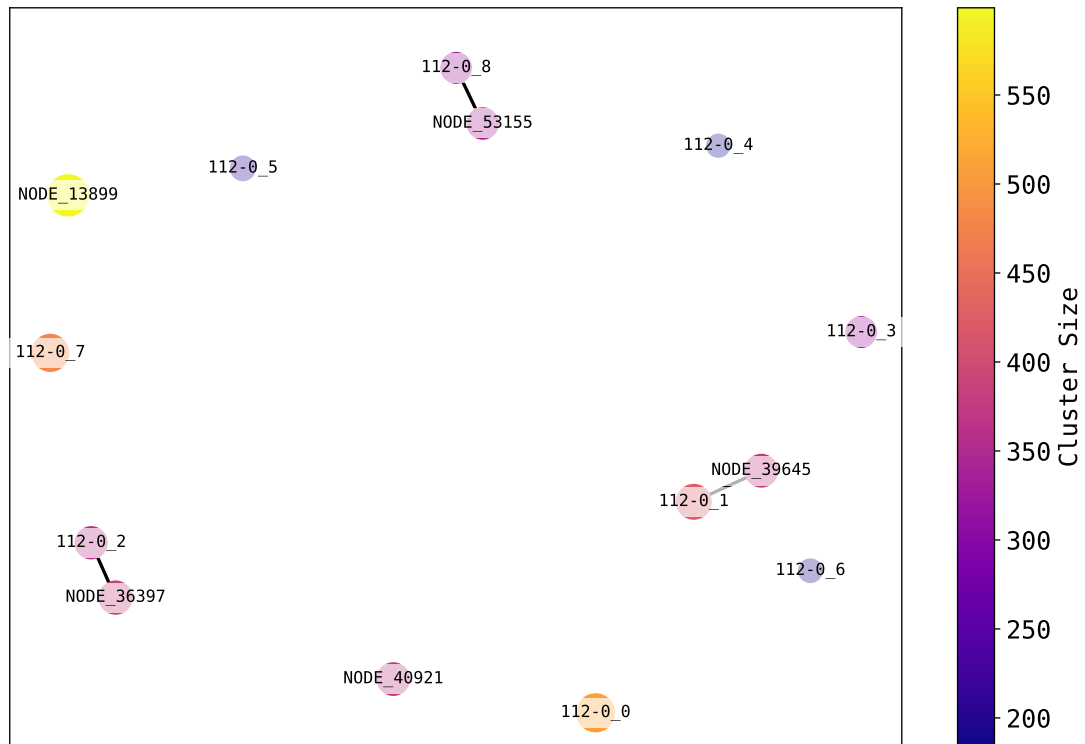


Fig. 7. Clusters evaluation for sample SRR15525307 and *Picornaviridae* family.

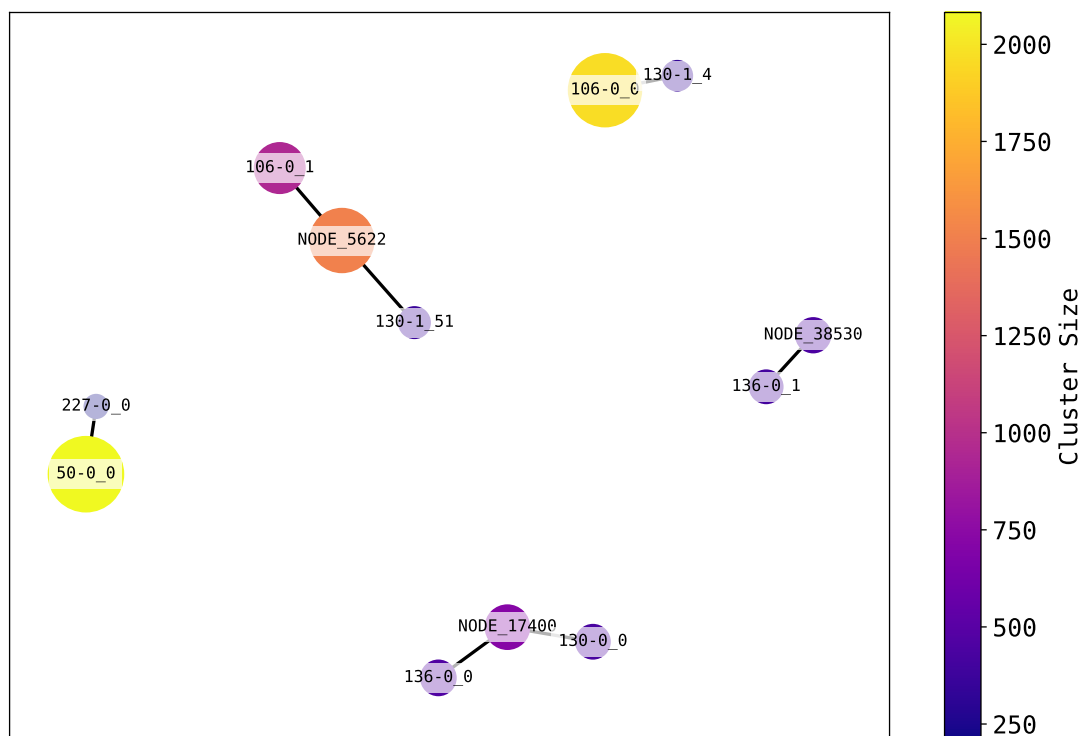


Fig. 8. Clusters evaluation for sample SRR15524477 and *Nodaviridae* family.

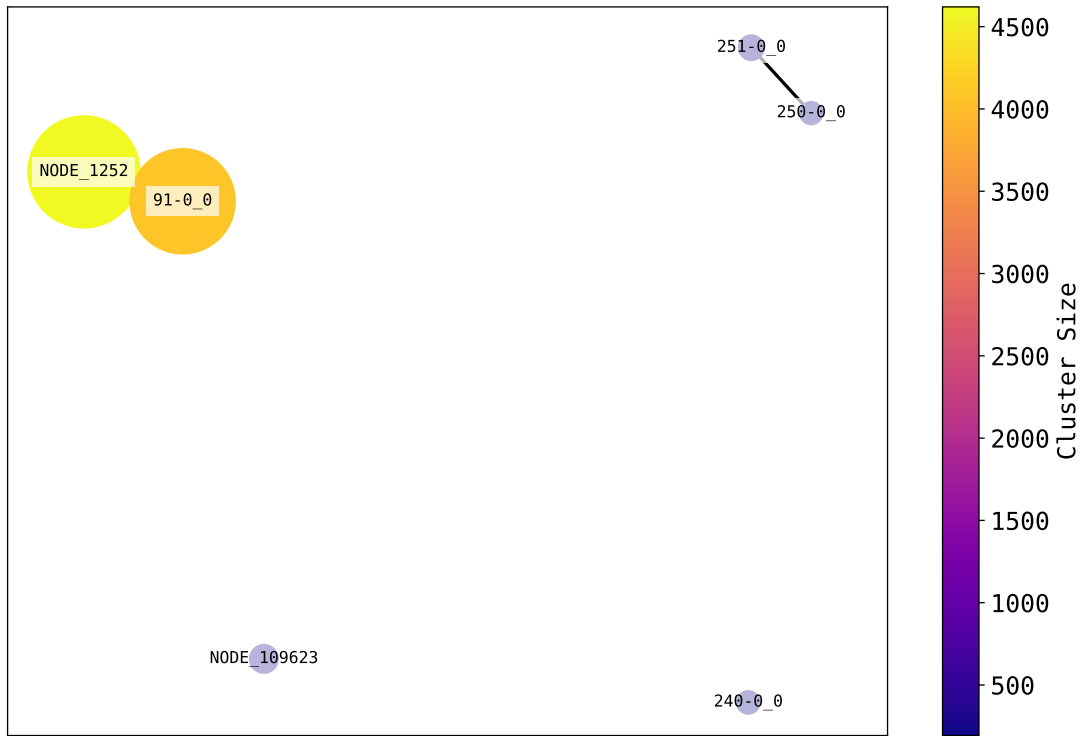


Fig. 9. Clusters evaluation for sample SRR15524477 and *Parvoviridae* family.

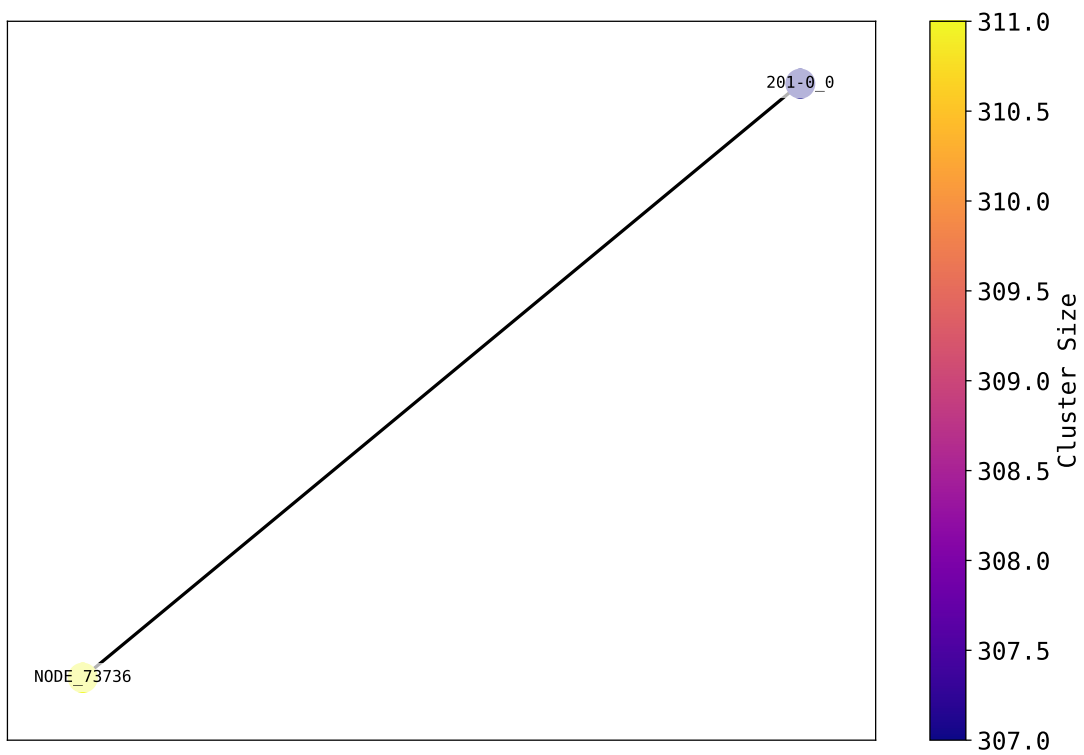


Fig. 10. Clusters evaluation for sample SRR15525307 and *Dicistroviridae* family.

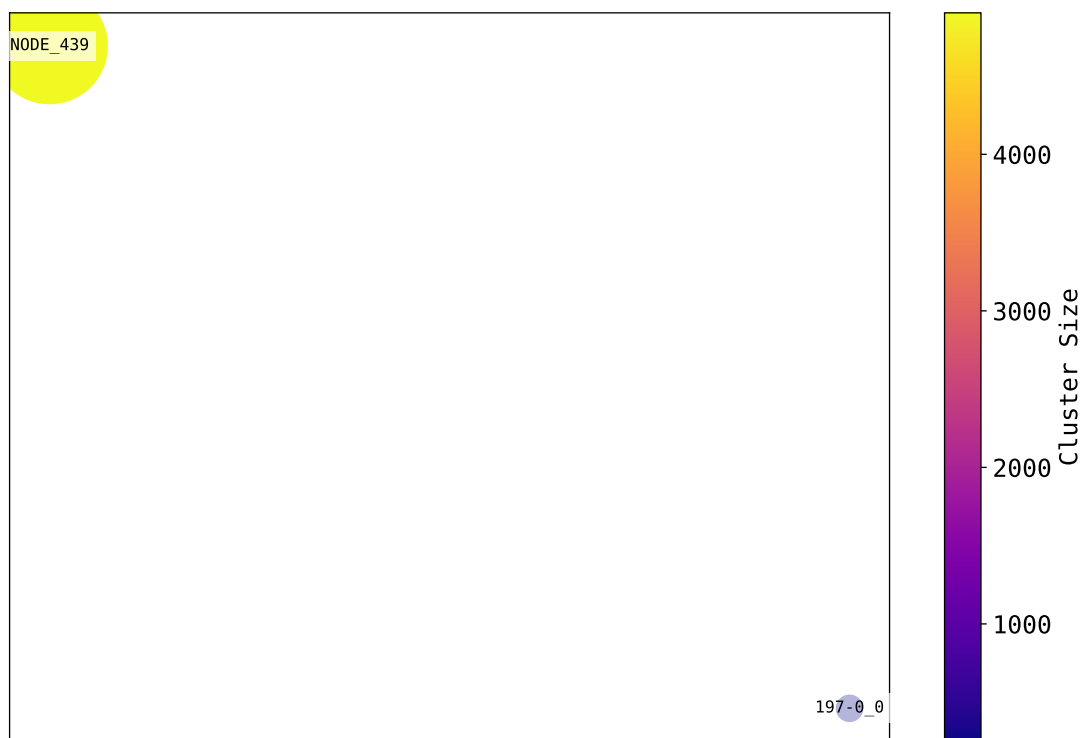


Fig. 11. Clusters evaluation for sample SRR15525307 and *Parvoviridae* family.

C. Pipeline resources consumption

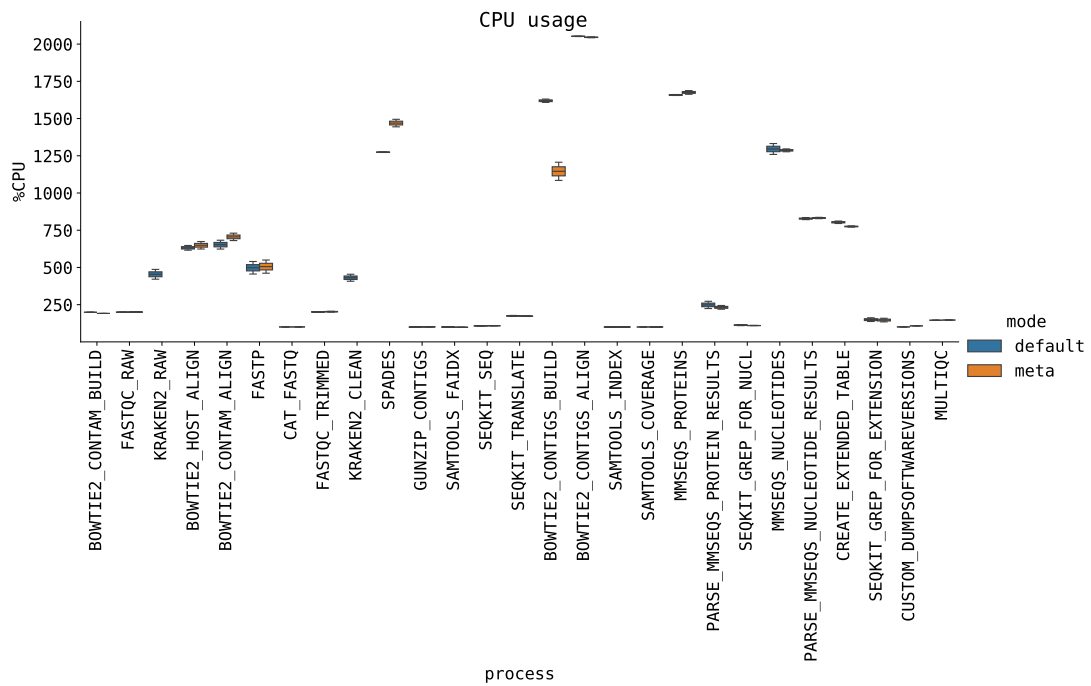


Fig. 12. CPU usage during the pipeline execution. The names of the processes are provided on the horizontal axis, while the vertical axis represents the percentage of CPU usage. Blue color represents `default` spades mode, orange color represents `meta` spades mode. Where color differentiation is not possible, the `default` spades mode is depicted on the left for the corresponding process. [Full size image](#).

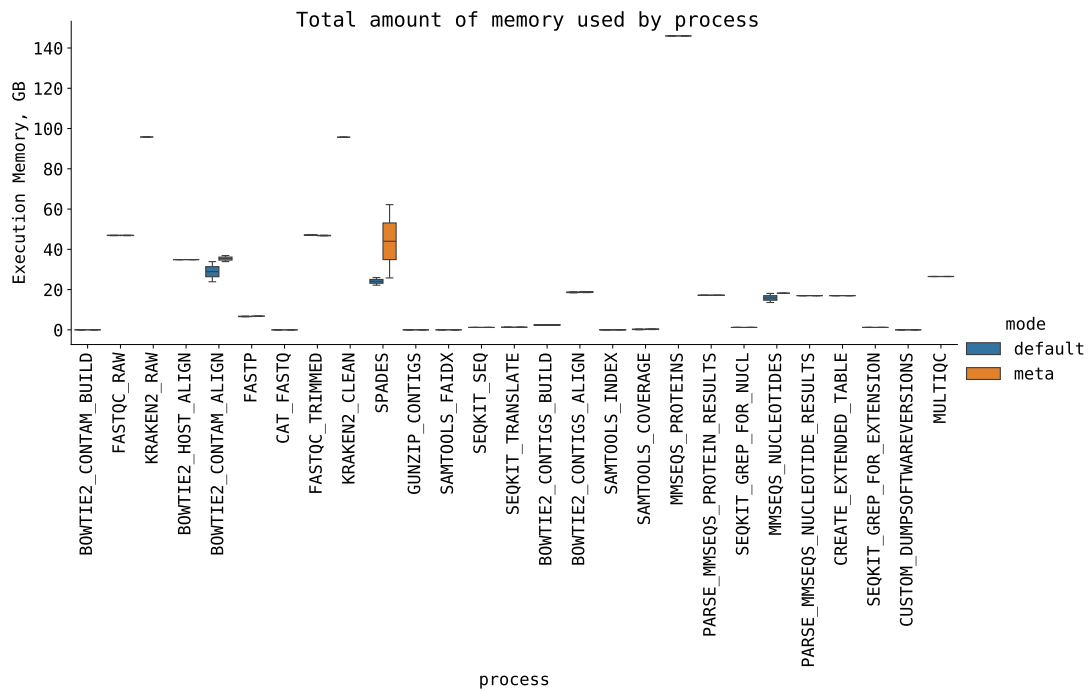


Fig. 13. RAM usage during the pipeline execution. The names of the processes are provided on the horizontal axis, while the vertical axis represents the percentage of RAM usage. Blue color represents default spades mode, orange color represents meta spades mode. Where color differentiation is not possible, the default spades mode is depicted on the left for the corresponding process. [Full size image](#).

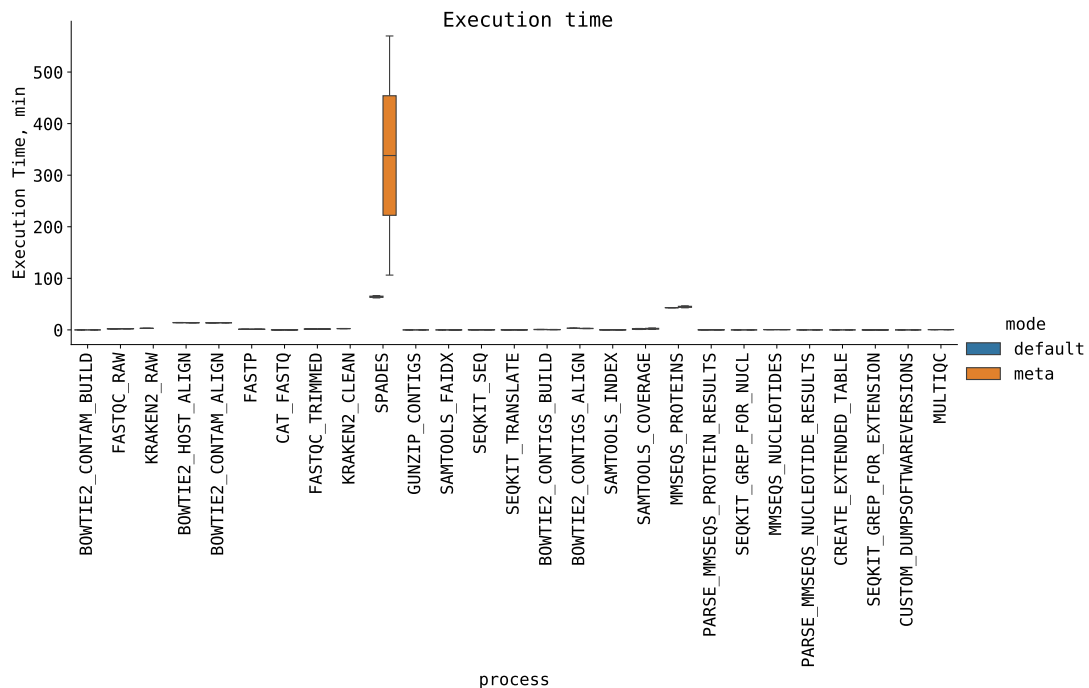


Fig. 14. Time span of processes during the pipeline execution. The names of the processes are provided on the horizontal axis, while the vertical axis represents the time span of the process. Blue color represents default spades mode, orange color represents meta spades mode. Where color differentiation is not possible, the default spades mode is depicted on the left for the corresponding process. [Full size image](#).

REFERENCES

1. Roychoudhury Sh., Das A., Sengupta P., Dutta S., Roychoudhury Sh., Choudhury A.P., Ahmed A.B.F., Bhattacharjee S., Slama P. Viral pandemics of the last four decades: pathophysiology, health impacts and perspectives. *International Journal of Environmental Research and Public Health*. 2020. V. 17. No. 24. Article No. 9411. doi: [10.3390/ijerph17249411](https://doi.org/10.3390/ijerph17249411)
2. Speranskaya A.S., Samoilov A.E., Korneenko E.V., Khabudaev K.V., Artyushin I.V., Yusefovich A.P., Dolgova A.S., Dedkov V.G. The characterization of complete genome of MERS-related coronavirus from bat (*Pipistrellus nathusii*) from Moscow region, prediction and analysis their Spike glycoprotein with DPP4 receptors of the different mammalian species. *Bioinformatics of Genome Regulation and Structure/Systems Biology (BGRS/SB-2022)*. 2022. P. 892–892.
3. Lau S.K.P., Li K.S.M., Tsang A.K.L., Lam C.S.F., Ahmed S., Chen H., Chan K.-H., Woo P.C.Y., Yuen K.-Y. Genetic characterization of Betacoronavirus lineage C viruses in bats reveals marked sequence divergence in the spike protein of pipistrellus bat coronavirus HKU5 in Japanese pipistrelle: implications for the origin of the novel Middle East respiratory syndrome coronavirus. *Journal of Virology*. 2013. V. 87. No. 15. P. 8638–8650. doi: [10.1128/jvi.01055-13](https://doi.org/10.1128/jvi.01055-13)
4. Speranskaya A.S., Artiushin I.V., Samoilov A.E., Korneenko E.V., Khabudaev K.V., Ilina E.N., Yusefovich A.P., Safonova M.V., Dolgova A.S., Gladkikh A.S., et al. Identification and genetic characterization of MERS-related coronavirus isolated from Nathusius' pipistrelle (*Pipistrellus nathusii*) near Zvenigorod (Moscow region, Russia). *International Journal of Environmental Research and Public Health*. 2023. V. 20. No. 4. Article No. 3702. doi: [10.3390/ijerph20043702](https://doi.org/10.3390/ijerph20043702)
5. Carroll D.S., Daszak P., Wolfe N.D., Gao G.F., Morel C.M., Morzaria S., Pablos-Méndez A., Tomori O.Y., Mazet J.A.K. The global virome project. *Science*. 2018. V. 359. No. 6378. P. 872–874.
6. Köster J., Rahmann S. Snakemake — a scalable bioinformatics workflow engine. *Bioinformatics*. 2012. V. 28. No. 19. P. 2520–2522. doi: [10.1093/bioinformatics/bts480](https://doi.org/10.1093/bioinformatics/bts480)
7. Di Tommaso P., Chatzou M., Floden E.W., Barja P.P., Palumbo E., Notredame C. Nextflow enables reproducible computational workflows. *Nature Biotechnology*. 2017. V. 35. No. 4. P. 316–319. doi: [10.1038/nbt.3820](https://doi.org/10.1038/nbt.3820)
8. Ewels P.A., Peltzer A., Fillinger S., Patel H., Alneberg J., Wilm A., Garcia M.U., Di Tommaso P., Nahnsen S. The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology*. 2020. V. 38. No. 3. P. 276–278. doi: [10.1038/s41587-020-0439-x](https://doi.org/10.1038/s41587-020-0439-x)
9. Patel H., Monzón S., Varona S., Espinosa-Carrasco J., Garcia M.U., Heuer M.L., Underwood A., Gabernet G., Ewels P., Miguel J., et al. *nf-core/viralrecon: nf-core/viralrecon v2. 6.0-Rhodium Raccoon*. 2023. doi: [10.5281/zenodo.7764938](https://doi.org/10.5281/zenodo.7764938)
10. Kieft K., Zhou Z., Anantharaman K. VIBRANT: automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences. *Microbiome*. 2020. V. 8. Article No. 90. doi: [10.1186/s40168-020-00867-0](https://doi.org/10.1186/s40168-020-00867-0)
11. Rampelli S., Soverini M., Turrone S., Quercia S., Biagi E., Brigidi P., Candela M. ViromeScan: a new tool for metagenomic viral community profiling. *BMC Genomics*. 2016. V. 17. Article No. 165. doi: [10.1186/s12864-016-2446-3](https://doi.org/10.1186/s12864-016-2446-3)

12. Vilsker M., Moosa Y., Nooij S., Fonseca V., Ghysens Y., Dumon K., Pauwels R., Alcantara L.C., Vanden Eynden E., Vandamme A.-M., et al. Genome Detective: an automated system for virus identification from high-throughput sequencing data. *Bioinformatics*. 2019. V. 35. No. 5. P. 871–873. doi: [10.1093/bioinformatics/bty695](https://doi.org/10.1093/bioinformatics/bty695)
13. Ren J., Song K., Deng C., Ahlgren N.A., Fuhrman J.A., Li Y., Xie X., Poplin R., Sun F. Identifying viruses from metagenomic data using deep learning. *Quantitative Biology*. 2020. V. 8. No. 1. P. 64–77. doi: [10.1007/s40484-019-0187-4](https://doi.org/10.1007/s40484-019-0187-4)
14. O’Leary N.A., Cox E., Holmes J.B., Anderson W.R., Falk R., Hem V., Tsuchiya M.T.N., Schuler G.D., Zhang X., Torcivia J., et al. Exploring and retrieving sequence and metadata for species across the tree of life with NCBI Datasets. *Scientific Data*. 2024. V. 11. Article No. 732. doi: [10.1038/s41597-024-03571-y](https://doi.org/10.1038/s41597-024-03571-y)
15. O’Leary N.A., Wright M.W., Brister J.R., Ciufu S., Haddad D., McVeigh R., Rajput B., Robbertse B., Smith-White B., Ako-Adjei D., et al. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*. 2016. V. 44. No. D1. P. D733–D745. doi: [10.1093/nar/gkv1189](https://doi.org/10.1093/nar/gkv1189)
16. Paez-Espino D., Eloie-Fadrosh E.A., Pavlopoulos G.A., Thomas A.D., Huntemann M., Mikhailova N., Rubin E., Ivanova N.N., Kyrpides N.C. Uncovering Earth’s virome. *Nature*. 2016. V. 536. No. 7617. P. 425–430. doi: [10.1038/nature19094](https://doi.org/10.1038/nature19094)
17. Guo J., Bolduc B., Zayed A.A., Varsani A., Dominguez-Huerta G., Delmont T.O., Pratama A.A., Gazitúa M.C., Vik D., Sullivan M.B., et al. VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses. *Microbiome*. 2021. V. 9. Article No. 1–13. doi: [10.1186/s40168-020-00990-y](https://doi.org/10.1186/s40168-020-00990-y)
18. Bukin Yu.S., Bondaryuk A.N., Butina T.V. Performance Analysis of Cross-Assembly of Metatranscriptomic Datasets in Viral Community Studies. *Mathematical Biology and Bioinformatics*. 2023. V. 18. No. 2. P. 418–433. doi: [10.17537/2023.18.418](https://doi.org/10.17537/2023.18.418)
19. Speranskaya A.S., Dorokhin A.V., Korneenko E.V., Chudinov I.K., Samoilov A.E., Kruskop S.V. A Novel Mastadenovirus from *Nyctalus noctula* Which Represents a Distinct Evolutionary Branch of Viruses from Bats in Europe. *Viruses*. 2024. V. 16. No. 8. Article No. 1207. doi: [10.3390/v16081207](https://doi.org/10.3390/v16081207)
20. Korneenko E.V., Samoilov A.E., Chudinov I.K., Butenko I.O., Sonets I.V., Artyushin I.V., Yusefovich A.P., Kruskop S.V., Sinitsyn S.O., Klyuchnikova E.O., et al. Alphacoronaviruses from bats captured in European Russia in 2015 and 2021 are closely related to those of Northern Europe. *Frontiers in Ecology and Evolution*. 2024. V. 12. doi: [10.3389/fevo.2024.1324605](https://doi.org/10.3389/fevo.2024.1324605)
21. Lukina-Gronskaya A.V., Chudinov I.K., Korneenko E.V., Mashkova S.D., Semashko T.A., Sinkova M.A., Penkin L.N., Litvinova E.M., Feoktistova N.Yu., Speranskaya A.S. Novel coronaviruses and mammarenaviruses of hedgehogs from Russia including the comparison of viral communities of hibernating and active specimens. *Frontiers in Veterinary Science*. 2024. V. 11. doi: [10.3389/fvets.2024.1486635](https://doi.org/10.3389/fvets.2024.1486635)
22. Langmead B., Salzberg S.L. Fast gapped-read alignment with Bowtie 2. *Nature Methods*. 2012. V. 9. No. 4. P. 357–359. doi: [10.1038/nmeth.1923](https://doi.org/10.1038/nmeth.1923)
23. *nf-core/modules*. URL: <https://nf-co.re/modules/> (accessed 06.05.2025).
24. Chen S., Zhou Y., Chen Y., Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics*. 2018. V. 34. No. 17. P. i884–i890. doi: [10.1093/bioinformatics/bty560](https://doi.org/10.1093/bioinformatics/bty560)
25. Andrews S. FastQC. Babraham Bioinformatics. URL: <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/> (accessed 06.05.2025).

26. *GNU Gzip: General file (de)compression*: Free Software Foundation, Inc. gunzip. URL: <https://www.gnu.org/software/gzip/manual/gzip.html> (accessed 06.05.2025).
27. Wood D.E., Lu J., Langmead B. Improved metagenomic analysis with Kraken 2. *Genome Biology*. 2019. V. 20. Article No. 257. doi: [10.1186/s13059-019-1891-0](https://doi.org/10.1186/s13059-019-1891-0)
28. MultiQC overview. *Seqera*. URL: <https://docs.seqera.io/multiqc> (accessed 06.05.2025).
29. Li H., Handsaker B., Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and SAMtools. *Bioinformatics*. 2009. V. 25. No. 16. P. 2078–2079. doi: [10.1093/bioinformatics/btp352](https://doi.org/10.1093/bioinformatics/btp352)
30. Shen W., Le S., Li Y., Hu F. SeqKit: a cross-platform and ultrafast toolkit for FASTA/Q file manipulation. *PloS One*. 2016. V. 11. No. 10. doi: [10.1371/journal.pone.0163962](https://doi.org/10.1371/journal.pone.0163962)
31. Shen W., Sipos B., Zhao L. SeqKit2: A Swiss army knife for sequence and alignment processing. *iMeta*. 2024. V. 3. No. 3. Article No. e191. doi: [10.1002/imt2.191](https://doi.org/10.1002/imt2.191)
32. Prjibelski A., Antipov D., Meleshko D., Lapidus A., Korobeynikov A. Using SPAdes de novo assembler. *Current Protocols in Bioinformatics*. 2020. V. 70. No. 1. P. e102. doi: [10.1002/cpbi.102](https://doi.org/10.1002/cpbi.102)
33. Van Rossum G., Drake F.L. Python 3 Reference Manual. *CreateSpace Independent Publishing Platform*. 2009.
34. Steinegger M., Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*. 2017. V. 35. P. 1026–1028. doi: [10.1038/nbt.3988](https://doi.org/10.1038/nbt.3988)
35. Steinegger M., Söding J. Clustering huge protein sequence sets in linear time. *Nature Communications*. 2018. V. 9. Article No. 2542. doi: [10.1038/s41467-018-04964-5](https://doi.org/10.1038/s41467-018-04964-5)
36. da Veiga Leprevost F., Grüning B., Alves A.S., Søndergaard H., Röst H.L., Uszkoreit J., Barsnes H., Vaudel M., Moreno P., Gatto L., et al. BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*. 2017. V. 33. No. 16. P. 2580–2582. doi: [10.1093/bioinformatics/btx192](https://doi.org/10.1093/bioinformatics/btx192)
37. Gruening B., Sallou O., Moreno P., da Veiga Leprevost F., Ménager H., Søndergaard D., Röst H., Sachsenberg T., O’connor B., Madeira F., et al. Recommendations for the packaging and containerizing of bioinformatics software. *F1000Research*. 2019. V. 7. doi: [10.12688/f1000research.15140.2](https://doi.org/10.12688/f1000research.15140.2)
38. Reback J., McKinney W., Jbrockmendel, Van Den Bossche J., Augspurger T., Cloud Ph., Gfyoung, Sinhrks, Klein A., Hawkins S., et al. *pandas-dev/pandas: Pandas 1.0.5*. 2020. doi: [10.5281/zenodo.3898987](https://doi.org/10.5281/zenodo.3898987)
39. Suzek B.E., Huang H., McGarvey P., Mazumder R., Wu C.H. UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*. 2007. V. 23. No. 10. P. 1282–1288. doi: [10.1093/bioinformatics/btm098](https://doi.org/10.1093/bioinformatics/btm098)
40. Sayers E.W., Bolton E.E., Brister J.R., Canese K., Chan J., Comeau D.C., Connor R., Funk K., Kelly C., Kim S., et al. Database resources of the national center for biotechnology information. *Nucleic Acids Research*. 2022. V. 50. No. D1. P. D20–D26. doi: [10.1093/nar/gkm1000](https://doi.org/10.1093/nar/gkm1000)
41. *NCBI. sra-tools*. URL: <https://github.com/ncbi/sra-tools> (accessed 06.05.2025).
42. Kluyver T., Ragan-Kelley B., Pérez F., Granger B., Bussonnier M., Frederic J., Kelley K., Hamrick J., Grout J., Corlay S., et al. Jupyter notebooks—a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. 2016. P. 87–90.

43. McKinney W. Data Structures for Statistical Computing in Python. In: *Proceedings of the 9th Python in Science Conference (SciPy 2010)*. Eds. Van Der Walt S.J. and Millman K.J. Austin, Texas, 2010. P. 56–61. doi: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a)
44. Hunter J.D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007. V. 9. No. 03. P. 90–95.
45. Hagberg A., Chult D.S., Swart P. Exploring Network Structure, Dynamics, and Function using NetworkX. In: *Proceedings of the 7th Python in Science Conference (SciPy 2008)*. Eds. Varoquaux G., Vaught T., Millman J. Pasadena, California, 2008. P. 11–16. doi: [10.25080/TCWV9851](https://doi.org/10.25080/TCWV9851)
46. Harris C.R., Millman K.J., Van Der Walt S.J., Gommers R., Virtanen P., Cournapeau D., Wieser E., Taylor J., Berg S., Smith N.J., et al. Array programming with NumPy. *Nature*. 2020. V. 585. P. 357–362. doi: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2)
47. Waskom M.L. Seaborn: statistical data visualization. *Journal of Open Source Software*. 2021. V. 6. No. 60. P. 3021.
48. *Inkscape project*. URL: <https://inkscape.org> (accessed 06.05.2025).
49. *PDFgear*. URL: <https://www.pdfgear.com> (accessed 06.05.2025).
50. Camacho C., Coulouris G., Avagyan V., Ma N., Papadopoulos J., Bealer K., Madden T.L. BLAST+: architecture and applications. *BMC Bioinformatics*. 2009. V. 10. doi: [10.1186/1471-2105-10-421](https://doi.org/10.1186/1471-2105-10-421)
51. Budkina A.Y., Korneenko E.V., Kotov I.A., Kiselev D.A., Artyushin I.V., Speranskaya A.S., Khafizov K., Akimkin V.G. Utilizing the VirIdAl pipeline to search for viruses in the metagenomic data of bat samples. *Viruses*. 2021. V. 13. No. 10. Article No. 2006. doi: [10.3390/v13102006](https://doi.org/10.3390/v13102006)
52. Huggett J.F., Benes V., Bustin S.A., Garson J.A., Harris K., Kammel M., Kubista M., McHugh T.D., Moran-Gilad J., Nolan T., et al. Cautionary note on contamination of reagents used for molecular detection of SARS-CoV-2. *Clinical Chemistry*. 2020. V. 66. No. 11. P. 1369–1372. doi: [10.1093/clinchem/hvaa214](https://doi.org/10.1093/clinchem/hvaa214)
53. Healy B., Khan A., Metezai H., Blyth I., Asad H. The impact of false positive COVID-19 results in an area of low prevalence. *Clinical Medicine*. 2021. V. 21. No. 1. P. e54–e56. doi: [10.7861/clinmed.2020-0839](https://doi.org/10.7861/clinmed.2020-0839)
54. Ahmed W., Simpson S.L., Bertsch P.M., Bibby K., Bivins A., Blackall L., Bofill-Mas S., Bosch A., Brandão J., Choi P.M., et al. Minimizing errors in RT-PCR detection and quantification of SARS-CoV-2 RNA for wastewater surveillance. *Science of the Total Environment*. 2022. V. 805. Article No. 149877. doi: [10.1016/j.scitotenv.2021.149877](https://doi.org/10.1016/j.scitotenv.2021.149877)
55. Wang C.Y.T., Buckley C., Bletchly C., Harris P., Whiley D. Contamination of SARS-CoV-2 RT-PCR probes at the oligonucleotide manufacturer. *Pathology*. 2020. V. 52. No. 7. P. 814–816. doi: [10.1016/j.pathol.2020.08.002](https://doi.org/10.1016/j.pathol.2020.08.002)
56. Volynkina A.S., Ryazanova A.G., Rusanova D.V., Kulichenko A.N. The Problem of DNA/RNA Contamination in the Laboratory during PCR Testing for COVID-19. *Public Health and Life Environment – PH&LE..* 2021. No. 7. P. 76–81 (in Russ.). doi: [10.35627/2219-5238/2021-29-7-76-81](https://doi.org/10.35627/2219-5238/2021-29-7-76-81)
57. Chen L.X., Banfield J.F. COBRA improves the completeness and contiguity of viral genomes assembled from metagenomes. *Nature Microbiology*. 2024. V. 9. P. 737–750. doi: [10.1038/s41564-023-01598-2](https://doi.org/10.1038/s41564-023-01598-2)
58. Aksamentov I., Roemer C., Hodcroft E.B., Neher R.A. Nextclade: clade assignment, mutation calling and quality control for viral genomes. *Journal of Open Source Software*. 2021. V. 6. No. 67. Article No. 3773. doi: [10.21105/joss.03773](https://doi.org/10.21105/joss.03773)

59. O'Toole A., Scher E., Underwood A., Jackson B., Hill V., McCrone J.T., Colquhoun R., Ruis C., Abu-Dahab K., Taylor B., et al. Assignment of epidemiological lineages in an emerging pandemic using the pangolin tool. *Virus Evolution*. 2021. V. 7. No. 2. Article No. veab064. doi: [10.1093/ve/veab064](https://doi.org/10.1093/ve/veab064)
60. Mistry J., Chuguransky S., Williams L., Qureshi M., Salazar G.A., Sonnhammer E.L.L., Tosatto S.E.C., Paladin L., Raj S., Richardson L.J., et al. Pfam: the protein families database in 2021. *Nucleic Acids Research*. 2021. V. 49. No. D1. P. D412–D419. doi: [10.1093/nar/gkaa913](https://doi.org/10.1093/nar/gkaa913)
61. Kans J. Entrez direct: E-utilities on the UNIX command line. *Entrez Programming Utilities Help [Internet]*. 2024. URL: <https://www.ncbi.nlm.nih.gov/books/NBK25499/> (accessed 06.05.2025).

Accepted 10.03.2025.

Revised 23.03.2025.

Published 14.05.2025.