Journal Title Here, 2022, pp. 1–5

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

nf-core/viralgenie: A Novel Pipeline For Untargeted Viral Genome Reconstruction

Joon Klaps[1,*] Philippe Lemey[1] and Liana Kafetzopoulou[1]

[1]Rega Institute for Medical Research Department of Microbiology, Immunology and Transplantation Department of Pharmaceutical and Pharmacological Sciences , KU Leuven, Herestraat 49, 3000, Leuven, Belgium
[*]Corresponding author. joon.klaps@kuleuven.be
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Motivation: Eukaryotic viruses present significant challenges in genome reconstruction and variant analysis due to their extensive diversity, quasi-species, absence of universal marker genes, and genome segmentation. While de novo assembly followed by reference database matching and consequently, read mapping is a common approach, manual execution of this workflow is extremely time-consuming, particularly due to the extensive reference verification and selection required. There is a critical need for an automated, scalable pipeline that can efficiently handle viral metagenomic analysis without manual intervention.
Results: Here, we present nf-core/viralgenie, a comprehensive viral metagenomic pipeline for untargeted genome reconstruction, and variant analysis of eukaryotic viruses. Viralgenie is implemented as a modular Nextflow workflow that processes metagenomic and hybridization capture enriched samples to automatically detect and assemble viral genomes, while also performing variant analysis. The pipeline features automated reference selection, quality control metrics, comprehensive documentation, and seamless integration with containerization technologies including Docker, Singularity, and Podman. We demonstrate its utility and accuracy through validation on both simulated and real datasets, showing robust performance across diverse viral families and sample types.
Availability: nf-core/viralgenie is freely available at https://github.com/nf-core/viralgenie with comprehensive documentation at https://nf-co.re/viralgenie.

Key words: viralgenie, bioinformatic pipeline, nextflow, viral metagenomics, viral assembly, viral variant analysis

Introduction

Reconstructing viral genomes from metagenomic sequencing data presents significant computational challenges, particularly for viruses that exhibit extensive genetic diversity with quasi-species formation. This diversity is further compounded by the prevalence of segmented genomes in many viral families, including influenza, rotavirus, and bunyaviruses, where individual segments may undergo independent evolutionary pressures and reassortment events.

Resulting in a challenging landscape for viral genome reconstruction. Typically, for accurate and complete viral genome reconstruction, manual curation of contigs and reference matching is required. This process is not only time-consuming making it impractical for large-scale studies or rapid response scenarios such as emerging viral outbreaks of unknown origin. The need for a more automated and scalable solution has become increasingly apparent, particularly in the context of public health surveillance and epidemiological research [NEED FOR CITATION].

To address these limitations, we developed nf-core/viralgenie, a comprehensive pipeline specifically designed for untargeted viral genome reconstruction. The pipeline implements an automated workflow that performs de novo assembly, reference matching through sequence clustering, and iterative refinement by read mapping and consensus calling to reconstruct viral genomes without prior knowledge of the target sequences. By integrating containerization technologies and following nf-core standards, viralgenie ensures reproducibility and scalability across diverse computational environments while maintaining the flexibility required for varied research applications.

Methods

The nf-core/viralgenie pipeline implements a comprehensive workflow for untargeted viral genome reconstruction and variant analysis, consisting of five major analytical subworkflows: preprocessing, metagenomic diversity assessment, assembly and polishing, variant analysis with iterative refinement, and consensus quality control. The pipeline is implemented

in Nextflow and follows nf-core standards [4], ensuring reproducibility and portability across computational environments through containerization with Docker, Singularity, or Conda.

Pipeline Overview and Installation

Viralgenie requires Nextflow and a container management system (Docker, Singularity, or Conda). The pipeline can be executed with minimal setup:

```
nextflow run nf-core/viralgenie \
    -profile docker \
    --input samplesheet.csv
```

Input data is provided through a samplesheet in CSV, TSV, YAML, or JSON format containing sample names and paths to FASTQ files. The pipeline supports both single-end and paired-end sequencing data, with optional support for Unique Molecular Identifiers (UMIs) and mapping constraints for reference-guided analysis.

Read Preprocessing

The preprocessing module performs quality control and filtering of raw sequencing reads through five sequential steps. Initial quality assessment is conducted using FastQC before and after each processing step to monitor data quality throughout the workflow.

Adapter trimming and read processing is performed using either fastp (default) or Trimmomatic, both of which provide comprehensive adapter removal and quality filtering capabilities. For libraries prepared with UMIs, PCR duplicate removal is implemented using HUMID, which supports both directional and maximum clustering methods for UMI-based deduplication. The directional method (default) accounts for expected PCR errors by grouping reads using the relationship: node A counts $\geq$ (2 $\times$ node B counts) - 1.

Read merging is performed when multiple sequencing runs exist for the same sample, concatenating R1 files with R1 and R2 files with R2, while maintaining separation between single-end and paired-end data. Complexity filtering, implemented through BBduk or prinseq++, removes low-complexity sequences containing repetitive elements that could produce spurious alignments during downstream analysis.

Host contamination removal is performed using Kraken2 [12] against a user-specified host genome database. The default database contains a subset of the human genome, though users are strongly encouraged to employ more comprehensive databases including complete host genomes, common sequencer contaminants, and bacterial genomes to ensure thorough decontamination.

Metagenomic Diversity Assessment

Taxonomic classification of processed reads is performed using two complementary approaches to maximize detection sensitivity across diverse viral families. Kaiju [8] performs protein-based classification using a Burrows-Wheeler transform search strategy against annotated protein-coding genes from microbial genomes, enabling detection of highly divergent sequences through amino acid conservation. Kraken2 [12] provides DNA-level classification using k-mer mapping to identify the lowest common ancestor (LCA) of genomes containing specific k-mers. Optional Bracken analysis can be enabled for abundance estimation, though viral abundance comparisons should be interpreted cautiously due to the absence of universal marker genes in viruses.

Results from both classifiers are visualized using Krona, which generates interactive multi-layered pie charts allowing hierarchical exploration of taxonomic diversity. This dual-classification approach compensates for the limitations of individual methods and provides comprehensive coverage of viral diversity in metagenomic samples.

Assembly and Polishing

The assembly module implements a multi-assembler approach followed by sophisticated clustering and scaffolding procedures. De novo assembly is performed using three complementary assemblers: SPAdes [1] (configured for RNA viral mode by default [7]), MEGAHIT, and Trinity. This multi-assembler strategy capitalizes on the distinct algorithmic strengths of each tool to maximize genome recovery across diverse viral families and coverage distributions.

Assembled contigs undergo extension using SSPACE Basic, which leverages paired-end read information to scaffold and extend initial assemblies. Coverage calculation is performed by mapping processed reads back to contigs using BWAmem2 (default), BWA, or Bowtie2, enabling identification and filtration of low-coverage assemblies that likely represent assembly artifacts.

Reference matching is conducted through BLASTn searches against a comprehensive reference sequence pool, with the default being the latest clustered Reference Viral Database (RVDB) [3]. The top five BLAST hits for each contig are retained and incorporated into subsequent clustering analysis, facilitating identification of related genomic segments and appropriate reference sequences for scaffolding.

Taxonomy-guided clustering employs a two-stage process to group related contigs. Initial pre-clustering uses taxonomic assignments from both Kraken2 and Kaiju to resolve classification inconsistencies and separate contigs by taxonomic identity. Subsequent nucleotide similarity clustering is performed using one of six available algorithms: CD-HIT-EST [6], VSEARCH [11], MMseqs-linclust, MMseqs-cluster, vRhyme, or Mash with network-based community detection. The choice of clustering method allows optimization for specific dataset characteristics, with CD-HIT-EST providing speed for smaller datasets and MMseqs variants offering scalability for larger analyses.

Final scaffolding maps all cluster members to their respective centroids using Minimap2 [5], followed by consensus calling with iVar to generate reference-assisted assemblies. Regions with zero coverage depth are optionally annotated using reference sequences to produce complete genome reconstructions.

Variant Analysis and Iterative Refinement

The variant calling module supports two distinct analytical pathways: external reference-based analysis and de novo assembly refinement. In external reference-based analysis, users provide reference genomes through mapping constraints, with automatic selection of the most appropriate references using Mash [10] k-mer distance calculations. This approach selects references sharing the highest number of k-mers with the sequencing reads, minimizing mapping bias for highly divergent viral sequences.

For de novo assembly refinement, the pipeline performs iterative improvement of initially assembled consensus genomes.

Each iteration maps reads back to the current consensus using BWAmem2, BWA, or Bowtie2, followed by variant calling and consensus generation. The default configuration performs two refinement iterations, though this is user-configurable.

Variant calling is implemented using either BCFtools or iVar, each offering distinct advantages for viral genomics applications. BCFtools provides higher precision through sophisticated statistical modeling but may miss low-frequency variants. iVar excels at detecting multiallelic sites and low-frequency variants, making it particularly suitable for viral quasi-species analysis. iVar also handles ambiguous nucleotides more effectively, representing multiallelic positions with IUPAC ambiguity codes rather than masking them.

Optional UMI-based deduplication can be performed using UMI-tools, while standard PCR duplicate removal utilizes Picard MarkDuplicates. Comprehensive mapping statistics are generated using samtools (flagstat, idxstats, stats), Picard CollectMultipleMetrics, and mosdepth for coverage analysis.

Variant filtering removes variants with insufficient depth or quality, with BCFtools implementing additional steps to handle multiallelic sites and merge SNPs with indels. The final consensus sequences incorporate high-quality variants while maintaining genomic completeness through reference-guided gap filling.

Consensus Quality Control and Annotation

Comprehensive quality assessment of reconstructed viral genomes is performed through multiple complementary analyses. QUAST provides standard assembly metrics including contig statistics, N50 values, and quantification of ambiguous bases, which serves as a primary indicator of consensus quality. CheckV [9] estimates genome completeness and contamination by comparison against a curated database of complete viral genomes, though completeness estimates for segmented viruses should be interpreted considering that CheckV calculates completeness based on concatenated segment lengths.

Functional annotation is performed using Prokka, which identifies coding sequences and assigns functional annotations. While originally designed for bacterial genomes, Prokka provides reasonable annotation for viral sequences, particularly when supplemented with custom viral protein databases such as prot-RVDB.

Consensus genomes undergo similarity analysis through BLASTn searches against the reference pool and MMseqs searches against comprehensive annotation databases such as Virosaurus [2]. MMseqs enables rapid tblastx-equivalent searches for highly divergent sequences while maintaining nucleotide database compatibility. The annotation pipeline extracts species identification, segment designation, expected host information, and additional metadata from the best database matches.

Multiple sequence alignment using MAFFT aligns final consensus genomes with their corresponding references and constituent de novo contigs, enabling assessment of assembly accuracy and identification of genomic variations. Variant functional annotation is performed using SnpEff, which predicts the biological impact of detected variants, including synonymous/non-synonymous classifications and amino acid changes.

All quality control metrics are integrated into interactive MultiQC reports, providing comprehensive visualization of pipeline results. Custom summary tables extract key metrics from each analysis tool, facilitating rapid assessment of reconstruction quality across multiple samples.

Results

The nf-core/viralgenie pipeline provides comprehensive outputs enabling thorough evaluation of viral genome reconstruction quality and downstream analysis preparation. This section describes the expected outputs, runtime characteristics, and parameter selection rationale that guide optimal pipeline performance.

Pipeline Performance and Runtime Metrics

Viralgenie demonstrates efficient computational performance across diverse sample types and scales. Runtime scales primarily with read depth and viral diversity, with typical processing times ranging from 2-6 hours for standard metagenomic samples (10-50 million reads) on modern compute clusters. The multi-assembler approach adds computational overhead compared to single-assembler pipelines but provides superior genome recovery, particularly for highly divergent or low-coverage viral sequences.

Memory requirements vary by analysis module, with assembly typically representing the most resource-intensive step. SPAdes requires the highest memory allocation (8-32 GB depending on dataset size), while MEGAHIT and Trinity offer more memory-efficient alternatives. The clustering and variant calling steps scale efficiently with read depth, maintaining reasonable resource requirements even for high-coverage datasets.

Pipeline scalability benefits from Nextflow's built-in parallelization capabilities, enabling concurrent processing of multiple samples and assembly methods. Resource allocation can be customized through configuration profiles, allowing optimization for different computational environments from local workstations to high-performance computing clusters.

Output Organization and Interpretation

Viralgenie generates a hierarchical output structure designed for intuitive navigation and comprehensive result interpretation. The primary MultiQC report serves as the central hub for quality assessment, presenting interactive visualizations of all major pipeline metrics. Custom summary tables within the MultiQC report extract key information from each analysis tool, enabling rapid identification of high-quality consensus genomes and potential issues requiring attention.

Consensus sequences are organized by sample and clustering results, with clear naming conventions indicating assembly methods and refinement iterations. Each consensus genome is accompanied by comprehensive metadata including quality metrics, annotation results, and mapping statistics. Intermediate files are preserved to enable detailed troubleshooting and alternative parameter exploration.

The `overview-tables` directory contains summarized results from all major analysis steps, providing convenient access to quantitative metrics for downstream analysis or publication. These tables include assembly statistics, taxonomy assignments, variant calling results, and quality control metrics in standardized formats compatible with common statistical software packages.

Default Parameter Selection and Tool Choices

The pipeline's default parameters reflect extensive benchmarking and optimization for viral metagenomic applications. Tool selection balances computational efficiency with analytical sensitivity, prioritizing methods that perform well across diverse viral families and sample types.

For clustering applications, the default CD-HIT-EST [6] algorithm with 85% similarity threshold provides an optimal balance between sensitivity and specificity for most viral datasets. This threshold effectively groups related genomic segments while maintaining separation of distinct viral strains. Alternative clustering methods are provided to accommodate specific research needs: VSEARCH [11] for enhanced accuracy, MMseqs variants for scalability, and Mash [10] for rapid approximate clustering.

Variant calling defaults favor iVar over BCFtools for consensus generation due to its superior handling of viral-specific challenges including multiallelic sites, low-frequency variants, and ambiguous base calling. However, BCFtools is employed for intermediate refinement steps where its conservative approach helps prevent error propagation during iterative improvement.

Database selections prioritize comprehensive coverage while maintaining computational tractability. The clustered RVDB serves as the default reference pool, providing broad viral representation while limiting computational requirements. For taxonomic classification, viral-specific databases are employed to maximize detection sensitivity for eukaryotic viruses while minimizing false positive assignments from bacterial or archaeal sequences.

Quality thresholds are conservatively set to ensure high-confidence results while accommodating the inherent challenges of viral genome reconstruction. Minimum read depth requirements (10x for variant calling), quality scores (Phred 20), and coverage thresholds are calibrated based on empirical performance across diverse viral families and sample preparation methods.

Discussion

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

This is an example for fourth level head - paragraph head

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Competing interests

No competing interest is declared.

Author contributions statement

Must include all authors, identified by initials, for example: S.R. and D.A. conceived the experiment(s), S.R. conducted the experiment(s), S.R. and D.A. analysed the results. S.R. and D.A. wrote and reviewed the manuscript.

Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions. This work is supported in part by funds from the National Science Foundation (NSF: # 1636933 and # 1920920).

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, Alexey V Pyshkin, Alexander V Sirotkin, Nikolay Vyahhi, Glenn Tesler, Max A Alekseyev, and Pavel A Pevzner. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. 19:455–477.
2. Anne Gleizes, Florian Laubscher, Nicolas Guex, Christian Iseli, Thomas Junier, Samuel Cordey, Jacques Fellay, Ioannis Xenarios, Laurent Kaiser, and Philippe Le Mercier. Virosaurus a reference to explore and capture virus genetic diversity. 12.
3. Norman Goodacre, Aisha Aljanahi, Subhiksha Nandakumar, Mike Mikailov, and Arifa S Khan. A reference viral database (RVDB) to enhance bioinformatics analysis of high-throughput sequencing for novel virus detection. 3.
4. Sabrina Krakau, Daniel Straub, Hadrien Gourlé, Gisela Gabernet, and Sven Nahnsen. nf-core/mag: a best-practice pipeline for metagenome hybrid assembly and binning. 4:lqac007.
5. Heng Li. Minimap2: pairwise alignment for nucleotide sequences. 34:3094–3100.
6. Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. 22:1658–1659.
7. Dmitry Meleshko, Iman Hajirasouliha, and Anton Korobeynikov. coronaSPAdes: from biosynthetic gene clusters to RNA viral assemblies. # RNAviral SPAdes: a assembler specific for RNA viruses. Solve problems of ¡b¿the bias RNA genomes have in evolution¡/b¿ (biases from the reverse transcription and polymerase chain reaction):¡b¿-¡/b¿Huge diversity ¡b¿- ¡/b¿Quasi species- Presence of Isoforms¡b¿MAIN GOAlL:¡/b¿ to remove possible variation due to quasispecies, strain variation and sequencing artifactsThe solution they have for is to modify some steps and include a HMM profile to simplify their de bruijn graph.
8. Peter Menzel, Kim Lee Ng, and Anders Krogh. Fast and sensitive taxonomic classification for metagenomics with kaiju. 7:11257.
9. Stephen Nayfach, Antonio Pedro Camargo, Frederik Schulz, Emiley Eloe-Fadrosh, Simon Roux, and Nikos C Kyrpides. CheckV assesses the quality and completeness of metagenome-assembled viral genomes. 39:578–585.
10. Brian D Ondov, Gabriel J Starrett, Anna Sappington, Aleksandra Kostic, Sergey Koren, Christopher B Buck, and Adam M Phillippy. Mash screen: high-throughput sequence containment estimation for genome discovery. 20:232.

11. Torbjørn Rognes, Tomáš Flouri, Ben Nichols, Christopher Quince, and Frédéric Mahé. VSEARCH: a versatile open source tool for metagenomics. 4:e2584.

12. Derrick E Wood, Jennifer Lu, and Ben Langmead. Improved metagenomic analysis with kraken 2. 20:257.