## Title

ViralWasm: a client-side user-friendly web application suite for viral genomics.

**Permalink**

https://escholarship.org/uc/item/2nc1g0kf

**Journal**

Bioinformatics, 40(1)

**Authors**

Ji, Daniel
Aboukhalil, Robert
Moshiri, Alexander

**Publication Date**

2024-01-02

**DOI**

10.1093/bioinformatics/btae018

Peer reviewed

*Bioinformatics*, 2024, **40(1)**, btae018
https://doi.org/10.1093/bioinformatics/btae018
Advance Access Publication Date: 10 January 2024
**Applications Note**

OXFORD

Genome analysis

# ViralWasm: a client-side user-friendly web application suite for viral genomics

**Daniel Ji[1], Robert Aboukhalil[2], Niema Moshiri [1,*]**

[1]Department of Computer Science & Engineering, UC San Diego, La Jolla, CA 92093, United States
[2]Chan Zuckerberg Initiative, Redwood City, CA 94063, United States

*Corresponding author. Department of Computer Science & Engineering, UC San Diego, 9500 Gilman Drive, La Jolla, CA 92093, United States. E-mail: niema@ucsd.edu (N.M.)

Associate Editor: Inanc Birol

## Abstract

**Motivation:** The genomic surveillance of viral pathogens such as SARS-CoV-2 and HIV-1 has been critical to modern epidemiology and public health, but the use of sequence analysis pipelines requires computational expertise, and web-based platforms require sending potentially sensitive raw sequence data to remote servers.

**Results:** We introduce ViralWasm, a user-friendly graphical web application suite for viral genomics. All ViralWasm tools utilize WebAssembly to execute the original command line tools client-side directly in the web browser without any user setup, with a cost of just 2-3x slowdown with respect to their command line counterparts.

**Availability and implementation:** The ViralWasm tool suite can be accessed at: https://niema-lab.github.io/ViralWasm

## 1 Introduction

The genomic surveillance of viral pathogens such as SARS-CoV-2 and HIV-1 has been critical to modern epidemiology and public health (Robishaw *et al*., 2021). Reconstructing viral genome sequences from samples collected from patients typically requires the use of sequence analysis pipelines such as iVar (Grubaugh *et al.* 2019), HAPHPIPE (Gibson *et al.* 2020), HAVoC (Truong Nguyen *et al.* 2021), V-pipe (Posada-Céspedes *et al.* 2021), VGEA (Oluniyi *et al.* 2021), and nf-core/viralrecon (Ewels *et al.* 2020). However, the use of such pipelines typically requires computational expertise to be able to install and execute command line tools. Alternatively, researchers are able to use web-based platforms such as Galaxy (The Galaxy Community et al. 2022) and Genome Detective (Vilsker *et al.* 2019), but use of these systems requires sending the raw sequence data (which may contain sensitive host contamination from the patient) to remote servers, which can be problematic with respect to patient privacy and HIPAA compliance (Banimfreg 2023).

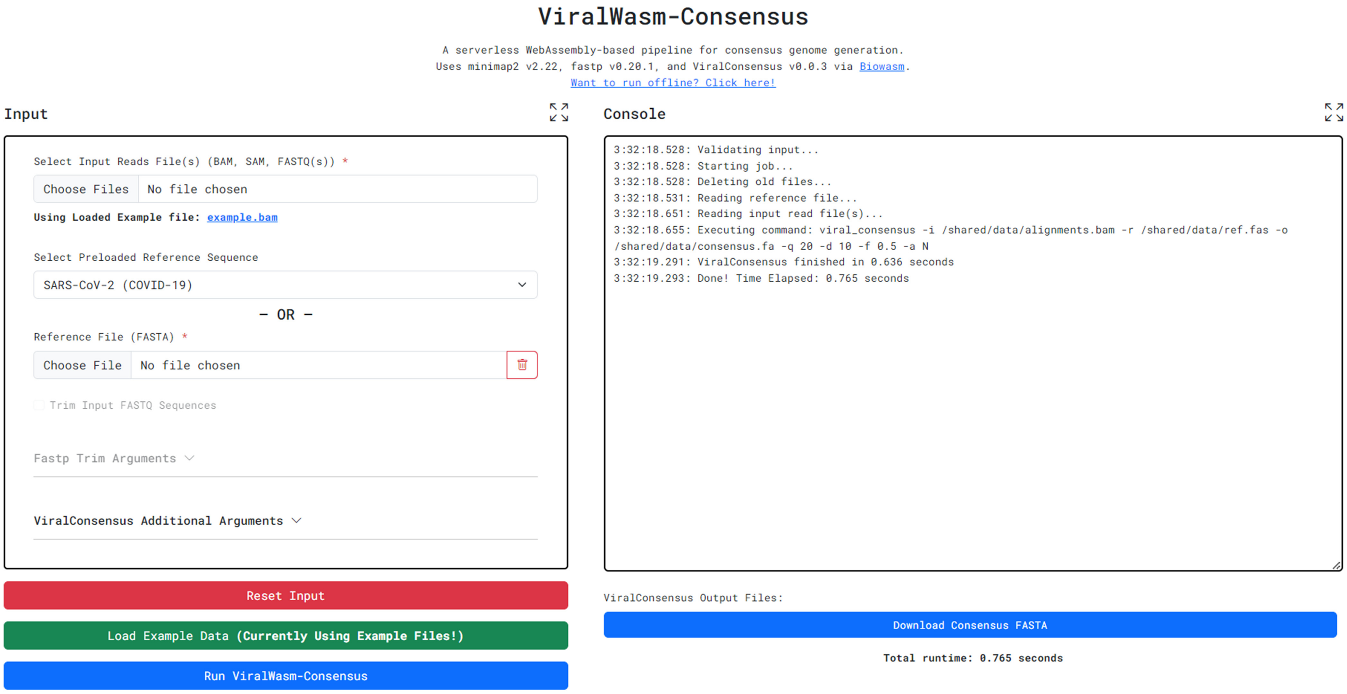
As web browsers have become increasingly sophisticated, tool developers have been given the ability to write complex web applications that are able to run client-side, directly in the user's web browser. One such advancement was the development of WebAssembly, which is a low-level bytecode into which native C/C++ code can be compiled and that aims to be safe, fast, portable, and compact (Haas *et al.* 2017), albeit with a slight performance cost with respect to native x86 code (Spies and Mock 2021). As a result, standard command line tools can be compiled into WebAssembly and incorporated into JavaScript web applications in which the WebAssembly modules are initialized once the user visits the website, without any user intervention, and executed directly in the web browser, with little to no modification of the original source code.

Here, we introduce ViralWasm, a client-side user-friendly graphical web application suite for viral genomics. Unlike existing viral genomics pipelines, ViralWasm tools require no setup whatsoever: once the user navigates to the ViralWasm website, all dependencies are loaded automatically as WebAssembly modules. Further, unlike existing pipelines, ViralWasm has an intuitive Graphical User Interface (GUI) with reasonable default values and examples. Lastly, and critically for sensitive patient data, ViralWasm runs completely client-side on the user's own machine (ie, the user's data are not sent anywhere), and it can even be run offline easily on Windows, Mac OS X, and Linux.

## 2 Methods

All ViralWasm tools utilize WebAssembly to execute the original command line tools client-side directly in the web browser. We use biowasm (biowasm.com) to load C and C++ tools, and we use Pyodide (pyodide.org) to load Python tools. Biowasm is a repository of recipes for compiling popular C/C++ bioinformatics tools to WebAssembly and provides pre-compiled WebAssembly binaries for use in web applications. The Aioli library provides JavaScript utilities for managing these WebAssembly modules and runs them in a background thread (WebWorkers) in the browser to maintain a responsive UI. As part of this project, we added recipes to

**Received:** 20 October 2023; **Editorial Decision:** 2 January 2024; **Accepted:** 9 January 2024



ViralWasm-Consensus

A serverless WebAssembly-based pipeline for consensus genome generation.
Uses minimap2 v2.22, fastp v0.20.1, and ViralConsensus v0.0.3 via Biowasm.
Want to run offline? Click here!

Input

Select Input Reads File(s) (BAM, SAM, FASTQ(s)) *
Choose Files No file chosen
**Using Loaded Example file:** example.bam

Select Preloaded Reference Sequence
SARS-CoV-2 (COVID-19)

– OR –

Reference File (FASTA) *
Choose File No file chosen

□ Trim Input FASTQ Sequences

Fastp Trim Arguments

ViralConsensus Additional Arguments

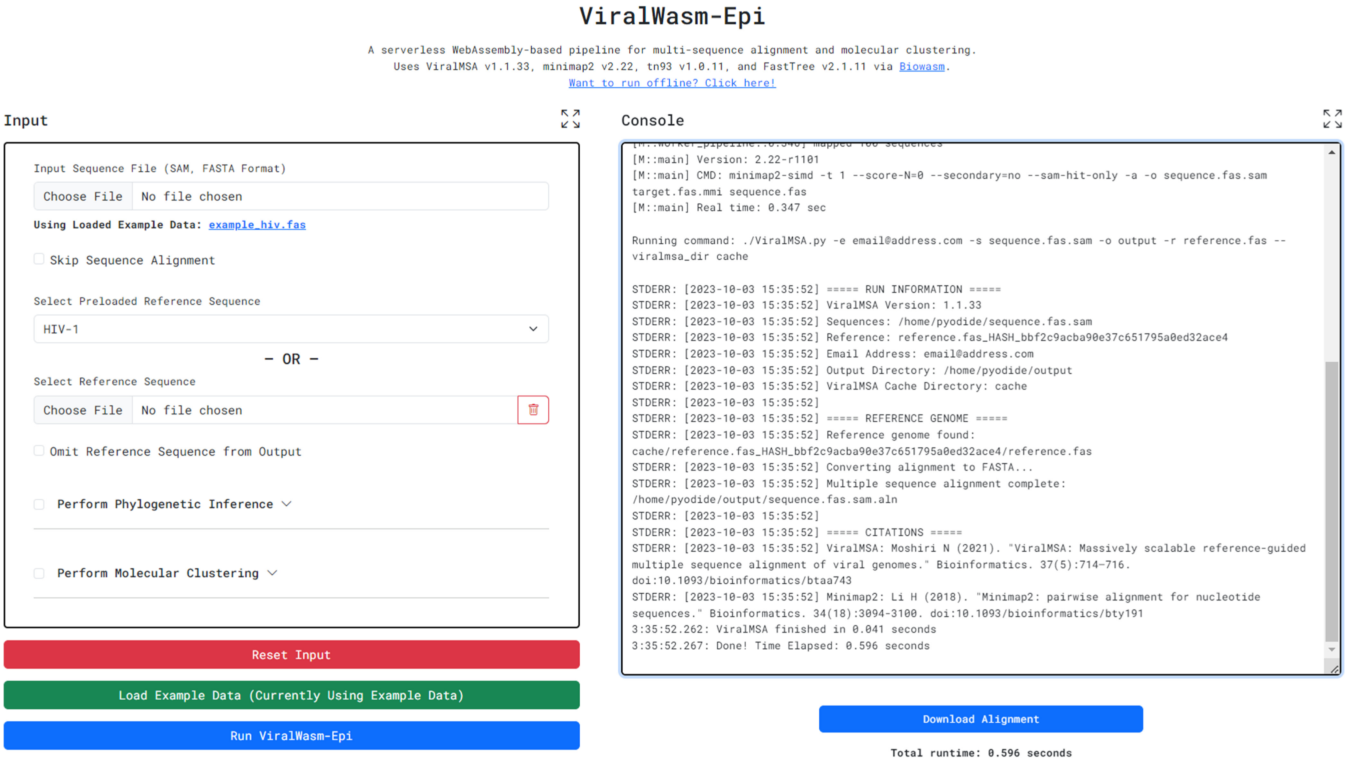
Console

```
3:32:18.528: Validating input...
3:32:18.528: Starting job...
3:32:18.528: Deleting old files...
3:32:18.531: Reading reference file...
3:32:18.651: Reading input read file(s)...
3:32:18.655: Executing command: viral_consensus -i /shared/data/alignments.bam -r /shared/data/ref.fas -o
/shared/data/consensus.fa -q 20 -d 10 -f 0.5 -a N
3:32:19.291: ViralConsensus finished in 0.636 seconds
3:32:19.293: Done! Time Elapsed: 0.765 seconds
```

ViralConsensus Output Files:

**Total runtime: 0.765 seconds**

**Figure 1.** ViralWasm-Consensus. Screenshot of an example ViralWasm-Consensus run.



ViralWasm-Epi

A serverless WebAssembly-based pipeline for multi-sequence alignment and molecular clustering.
Uses ViralMSA v1.1.33, minimap2 v2.22, tn93 v1.0.11, and FastTree v2.1.11 via Biowasm.
Want to run offline? Click here!

Input

Input Sequence File (SAM, FASTA Format)
Choose File No file chosen
**Using Loaded Example Data:** example_hiv.fas

□ Skip Sequence Alignment

Select Preloaded Reference Sequence
HIV-1

– OR –

Select Reference Sequence
Choose File No file chosen

□ Omit Reference Sequence from Output

□ Perform Phylogenetic Inference

□ Perform Molecular Clustering

Console

```
[M::main] Version: 2.22-r1101
[M::main] CMD: minimap2-simd -t 1 --score-N=0 --secondary=no --sam-hit-only -a -o sequence.fas.sam
target.fas.mmi sequence.fas
[M::main] Real time: 0.347 sec

Running command: ./ViralMSA.py -e email@address.com -s sequence.fas.sam -o output -r reference.fas --
viralmsa_dir cache

STDERR: [2023-10-03 15:35:52] ===== RUN INFORMATION =====
STDERR: [2023-10-03 15:35:52] ViralMSA Version: 1.1.33
STDERR: [2023-10-03 15:35:52] Sequences: /home/pyodide/sequence.fas.sam
STDERR: [2023-10-03 15:35:52] Reference: reference.fas_HASH_bbf2c9acba90e37c651795a0ed32ace4
STDERR: [2023-10-03 15:35:52] Email Address: email@address.com
STDERR: [2023-10-03 15:35:52] Output Directory: /home/pyodide/output
STDERR: [2023-10-03 15:35:52] ViralMSA Cache Directory: cache
STDERR: [2023-10-03 15:35:52]
STDERR: [2023-10-03 15:35:52] ===== REFERENCE GENOME =====
STDERR: [2023-10-03 15:35:52] Reference genome found:
cache/reference.fas_HASH_bbf2c9acba90e37c651795a0ed32ace4/reference.fas
STDERR: [2023-10-03 15:35:52] Converting alignment to FASTA...
STDERR: [2023-10-03 15:35:52] Multiple sequence alignment complete:
/home/pyodide/output/sequence.fas.sam.aln
STDERR: [2023-10-03 15:35:52]
STDERR: [2023-10-03 15:35:52] ===== CITATIONS =====
STDERR: [2023-10-03 15:35:52] ViralMSA: Moshiri N (2021). "ViralMSA: Massively scalable reference-guided
multiple sequence alignment of viral genomes." Bioinformatics. 37(5):714–716.
doi:10.1093/bioinformatics/btaa743
STDERR: [2023-10-03 15:35:52] Minimap2: Li H (2018). "Minimap2: pairwise alignment for nucleotide
sequences." Bioinformatics. 34(18):3094-3100. doi:10.1093/bioinformatics/bty191
3:35:52.262: ViralMSA finished in 0.041 seconds
3:35:52.267: Done! Time Elapsed: 0.596 seconds
```

**Total runtime: 0.596 seconds**

**Figure 2.** ViralWasm-Epi. Screenshot of an example ViralWasm-Epi run.

biowasm for tn93, ViralConsensus, and FastTree so that other developers can use these tools in their own web applications as well. The GUIs of all tools utilize Sassy CSS and Bootstrap for aesthetics. All testing and benchmarking is performed in a "Continuous Integration/Continuous Delivery" fashion using GitHub Actions, running Playwright tests on an Ubuntu 22.04 environment. The web applications themselves are deployed using GitHub Pages. We also utilize the Pako Javascript package in order to compress user-loaded files to memory before running the pipelines, and we use the Marked JavaScript package to parse Markdown for display on the websites. To run the web applications in offline mode, we provide a Python script that starts a local multithreaded HTTP server.

## 3 Results

The ViralWasm application suite consists of a main landing page (niema-lab.github.io/ViralWasm) that navigates users to the core pipelines (ViralWasm-Consensus and ViralWasm-Epi) and standalone tool web applications. For the sake of reproducibility, ViralWasm web applications specify the versions of all Bioinformatics tools they utilize, and all web applications can be downloaded as a zip archive for local deployment on the user's machine. For the sake of security, these locally downloaded archives can be deployed completely offline. For the sake of education, ViralWasm tools print to the log the exact commands that would be executed to run each tool from the command line, meaning ViralWasm can be used as an instructional tool to learn how to run Bioinformatics workflows from the command line.

### 3.1 ViralWasm-Consensus

ViralWasm-Consensus (niema-lab.github.io/ViralWasm-Consensus) is a pipeline for consensus sequence generation (Fig. 1). As input, users provide (i) viral reads in the FASTQ, SAM, or BAM file formats and (ii) a reference genome in the FASTA format or select one from a preloaded set of popular viral reference genomes.

If the user provides a FASTQ file as input, the reads are mapped against the user-selected reference genome using Minimap2 (Li 2018) using its default mapping parameters. Optionally, users can choose to first trim the reads using fastp (Chen *et al.* 2018). For the sake of simplicity, we currently only support trimming a user-specified number of bases from the front and/or tail of reads, polyG tail trimming, and 3' polyX trimming, but any fastp parameters can be added in the future.

A consensus viral genome sequence is then constructed from the SAM/BAM using ViralConsensus (Moshiri 2023). Sensible parameter values are selected by default, but all ViralConsensus parameters can be overridden as desired by the user (e.g. to increase or decrease strictness).

Once the execution has completed, the user can download the resulting consensus sequence in the FASTA file format, and the user can also optionally download any other output files that were produced during the execution of the pipeline (e.g. base and insertion counts produced by ViralConsensus, the SAM file produced by Minimap2, and the trimmed FASTQ files produced by fastp).

### 3.2 ViralWasm-Epi

ViralWasm-Epi (niema-lab.github.io/ViralWasm-Epi) is a pipeline for viral molecular epidemiology (Fig. 2). As input, users provide (i) viral genome sequences in the FASTA file format, (ii) a reference genome in the FASTA format or select one from a preloaded set of popular viral reference genomes, and optionally (iii) a plain-text file containing sample collection dates.

If the user provides an unaligned FASTA file as input, multiple sequence alignment is performed using ViralMSA (Moshiri 2021). The user can then choose to perform phylogenetic inference using FastTree 2 (Price *et al.* 2010), phylogenetic rooting and dating using LSD2 (To *et al.* 2016), and/or single-linkage genetic distance molecular clustering under the Tamura-Nei 93 (TN93) substitution model (Tamura and Nei 1993) using the tn93 tool from HIV-TRACE (Kosakovsky Pond *et al.* 2018). All FastTree 2, LSD2, and tn93 parameters can be adjusted from their default values by the user as desired.

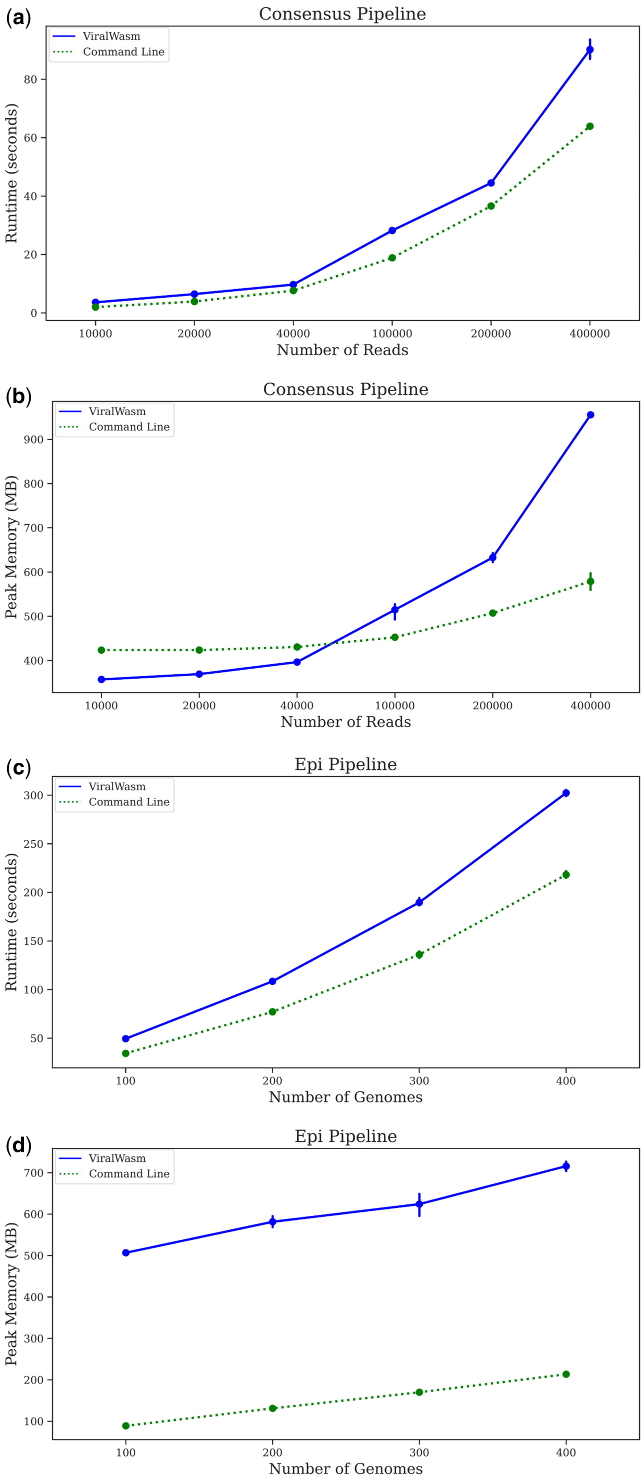


**Figure 3.** Benchmark results. Runtime and peak memory usage for ViralWasm and its command line counterparts for the Consensus (a, b) and Epi (c, d) pipelines for various dataset sizes.

Once the execution has completed, the user can download any output files (e.g. the FASTA file produced by ViralMSA, the distances TSV file produced by tn93, the clusters TSV file produced by ViralWasm-Epi, and the Newick tree files produced by FastTree 2 and LSD2).

### 3.3 Standalone web applications

In addition to ViralWasm-Consensus and ViralWasm-Epi, which are intended to be complete end-to-end pipelines for their respective analyses, the ViralWasm application suite also provides standalone web applications for each individual tool utilized by either pipeline, as well as for other Bioinformatics tools that may be of interest for viral analyses. Currently, ViralWasm contains standalone web applications for the tn93 tool from HIV-TRACE, ViralConsensus, and ViralMSA.

### 3.4 Benchmarking

ViralWasm-Consensus and ViralWasm-Epi were benchmarked against their exact command line counterparts. ViralWasm-Consensus was benchmarked using a SARS-CoV-2 Illumina amplicon sequencing dataset that was subsampled to various numbers of reads (Moshiri *et al.* 2022), and ViralWasm-Epi was benchmarked using a dataset of full HIV-1 genome sequences obtained from NCBI Virus. All benchmarks were performed via GitHub Actions using the default configuration, which runs in an Ubuntu 22.04 environment running on Azure Standard_DS2_v2 virtual machines with 2 Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz CPUs and 7GB of RAM. Command line tools were compiled and installed from source, and ViralWasm tools were executed using Playwright (playwright.dev).

As can be seen in Fig. 3, the ViralWasm-Consensus and ViralWasm-Epi pipelines are slightly slower and have higher peak memory usage than their command line counterparts. This is expected: by compiling the tools into WebAssembly rather than native code, ViralWasm gains convenience and ease-of-use at the expense of performance. However, both pipelines still run quite quickly with reasonable memory usage on fairly large datasets. ViralWasm-Consensus took roughly 90 s and used roughly 1 GB of memory to run on a dataset containing 400 000 SARS-CoV-2 reads (roughly 2000× coverage), well above what is needed for accurate consensus sequence calls from amplicon sequencing data (Moshiri 2023). ViralWasm-Epi took roughly 300 s and used roughly 700 MB of memory to run on a dataset containing 400 complete HIV-1 genome sequences, which is a reasonable size for real-world molecular epidemiological studies (Little *et al.* 2014).

## Acknowledgements

We would like to thank Rob Knight, Kristian Andersen, Louise Laurent, Katie Fisch, Tajana Rosing, Sean Eilert, Ameen Akel, Justin Eno, and Ken Curewitz for fruitful conversations.

## Conflict of interest

None declared.

## Funding

This work was supported by UC San Diego Faculty Research Funds and CDC contract 75D30122C14843.

## References

Banimfreg BH. A comprehensive review and conceptual framework for cloud computing adoption in bioinformatics. *Healthc Anal* 2023;**3**:100190.

Chen S, Zhou Y, Chen Y *et al.* fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics* 2018;**34**:i884–90.

Ewels PA, Peltzer A, Fillinger S *et al.* The nf-core framework for community-curated bioinformatics pipelines. *Nat Biotechnol* 2020; **38**:276–8.

Gibson KM, Steiner MC, Rentia U *et al.* Validation of variant assembly using HAPHPIPE with Next-Generation sequence data from viruses. *Viruses* 2020;**12**:758.

Grubaugh ND, Gangavarapu K, Quick J *et al.* An amplicon-based sequencing framework for accurately measuring intrahost virus diversity using PrimalSeq and iVar. *Genome Biol* 2019;**20**:8.

Haas A, Rossberg A, Schuff DL *et al.* Bringing the web up to speed with WebAssembly. *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*. Barcelona Spain: ACM2017. 185–200.

Kosakovsky Pond SL, Weaver S, Leigh Brown AJ *et al.* HIV-TRACE (TRAnsmission cluster engine): a tool for large scale molecular epidemiology of HIV-1 and other rapidly evolving pathogens. *Mol Biol Evol* 2018;**35**:1812–9.

Li H. Minimap2: pairwise alignment for nucleotide sequences. Birol I (ed.). *Bioinformatics* 2018;**34**:3094–100.

Little SJ, Kosakovsky Pond SL, Anderson CM *et al.* Using HIV networks to inform real time prevention interventions. Harrigan PR (ed.). *PLoS ONE* 2014;**9**:e98443.

Moshiri N. ViralMSA: massively scalable reference-guided multiple sequence alignment of viral genomes. Robinson P (ed.). *Bioinformatics* 2021;**37**:714–6.

Moshiri N. ViralConsensus: a fast and memory-efficient tool for calling viral consensus genome sequences directly from read alignment data. Marschall T (ed.). *Bioinformatics* 2023;**39**:btad317.

Moshiri N, Fisch KM, Birmingham A *et al.* The ViReflow pipeline enables user friendly large scale viral consensus genome reconstruction. *Sci Rep* 2022;**12**:5077.

Oluniyi PE, Ajogbasile F, Oguzie J *et al.* VGEA: an RNA viral assembly toolkit. *PeerJ* 2021;**9**:e12129.

Posada-Céspedes S, Seifert D, Topolsky I *et al.* V-pipe: a computational pipeline for assessing viral genetic diversity from high-throughput data. Xu J (ed.). *Bioinformatics* 2021;**37**:1673–80.

Price MN, Dehal PS, Arkin AP. FastTree 2—approximately maximum-likelihood trees for large alignments. Poon AFY (ed.). *PLoS ONE* 2010;**5**:e9490.

Robishaw JD, Alter SM, Solano JJ *et al.* Genomic surveillance to combat COVID-19: challenges and opportunities. *The Lancet Microbe* 2021;**2**:e481–4.

Spies B, Mock M. An Evaluation of WebAssembly in non-web environments. *2021 XLVII Latin American Computing Conference (CLEI)*. Cartago, Costa Rica: IEEE, 2021, 1–10.

Tamura K, Nei M. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol Biol Evol* 1993;**10**:512–26.

The Galaxy Community, Afgan E, Nekrutenko A, *et al.* The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic Acids Res* 2022;**50**:W345–51.

To T-H, Jung M, Lycett S *et al.* Fast dating using least-squares criteria and algorithms. *Syst Biol* 2016;**65**:82–97.

Truong Nguyen PT, Plyusnin I, Sironen T *et al.* HAVoC, a bioinformatic pipeline for reference-based consensus assembly and lineage assignment for SARS-CoV-2 sequences. *BMC Bioinform* 2021;**22**:373.

Vilsker M, Moosa Y, Nooij S *et al.* Genome detective: an automated system for virus identification from high-throughput sequencing data. Birol I (ed.). *Bioinformatics* 2019;**35**:871–3.