SOFTWARE TOOL ARTICLE

REVISED Reproducibly sampling SARS-CoV-2 genomes across time, geography, and viral diversity [version 2; peer review: 2 approved]

Evan Bolyen[1,2], Matthew R. Dillon[1], Nicholas A. Bokulich[3], Jason T. Ladner[4], Brendan B. Larsen[5], Crystal M. Hepp[2,4], Darrin Lemmer[6], Jason W. Sahl[4,7], Andrew Sanchez[1], Chris Holdgraf[8], Chris Sewell[9], Aakash G. Choudhury[10], John Stachurski[10], Matthew McKay[10], Anthony Simard[1], David M. Engelthaler[6], Michael Worobey[5], Paul Keim[4,6,7], J. Gregory Caporaso[1,7]

[1]Center for Applied Microbiome Science, Pathogen and Microbiome Institute, Northern Arizona University, Flagstaff, AZ, USA
[2]School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ, USA
[3]Laboratory of Food Systems Biotechnology, Institute of Food, Nutrition and Health, ETH Zurich, Switzerland
[4]Pathogen and Microbiome Institute, Northern Arizona University, Flagstaff, AZ, USA
[5]Department of Ecology and Evolutionary Biology, University of Arizona, Tucson, AZ, USA
[6]Pathogen and Microbiome Division, Translational Genomics Research Institute, Flagstaff, AZ, USA
[7]Department of Biological Sciences, Northern Arizona University, Flagstaff, AZ, USA
[8]Department of Statistics, University of California at Berkeley, Berkeley, CA, USA
[9]Theory and Simulation of Materials, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[10]Research School of Economics, Australian National University, ACT, Australia

v2 **First published:** 29 Jun 2020, **9**:657
https://doi.org/10.12688/f1000research.24751.1
Latest published: 28 Oct 2020, **9**:657
https://doi.org/10.12688/f1000research.24751.2

Abstract

The COVID-19 pandemic has led to a rapid accumulation of SARS-CoV-2 genomes, enabling genomic epidemiology on local and global scales. Collections of genomes from resources such as GISAID must be subsampled to enable computationally feasible phylogenetic and other analyses. We present genome-sampler, a software package that supports sampling collections of viral genomes across multiple axes including time of genome isolation, location of genome isolation, and viral diversity. The software is modular in design so that these or future sampling approaches can be applied independently and combined (or replaced with a random sampling approach) to facilitate custom workflows and benchmarking. genome-sampler is written as a QIIME 2 plugin, ensuring that its application is fully reproducible through QIIME 2's unique retrospective data provenance tracking system. genome-sampler can be installed in a conda environment on macOS or Linux systems. A complete default pipeline is available through a Snakemake workflow, so subsampling can be achieved using a single command. genome-sampler is open source, free for all to use, and available at https://caporasolab.us/genome-sampler. We

Open Peer Review

Reviewer Status ✓ ✓

	Invited Reviewers	
	1	**2**
version 2 (revision) 28 Oct 2020		✓ report
version 1 29 Jun 2020	✓ report	? report

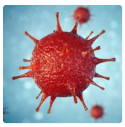
1. **James Hadfield**, Fred Hutchinson Cancer Research Center, Seattle, USA

2. **C. Titus Brown**, University of California, Davis, Davis, USA

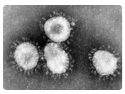
hope that this will facilitate SARS-CoV-2 research and support evaluation of viral genome sampling approaches for genomic epidemiology.

Keywords
SARS-CoV-2, genome-sampler, QIIME 2, bioinformatics, genomics

Any reports and responses or comments on the article can be found at the end of the article.



This article is included in the Disease Outbreaks gateway.



This article is included in the Coronavirus collection.

Corresponding author: J. Gregory Caporaso (greg.caporaso@nau.edu)

Author roles: Bolyen E: Conceptualization, Software, Writing – Review & Editing; **Dillon MR**: Resources, Software, Visualization, Writing – Review & Editing; **Bokulich NA**: Software, Writing – Review & Editing; **Ladner JT**: Conceptualization; **Larsen BB**: Conceptualization; **Hepp CM**: Conceptualization; **Lemmer D**: Conceptualization, Data Curation; **Sahl JW**: Conceptualization, Data Curation, Writing – Review & Editing; **Sanchez A**: Software, Visualization; **Holdgraf C**: Software, Writing – Review & Editing; **Sewell C**: Software; **Choudhury AG**: Software; **Stachurski J**: Software; **McKay M**: Software; **Simard A**: Software; **Engelthaler DM**: Conceptualization, Supervision, Writing – Review & Editing; **Worobey M**: Conceptualization, Supervision; **Keim P**: Conceptualization, Supervision; **Caporaso JG**: Conceptualization, Funding Acquisition, Project Administration, Software, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: Our software development and documentation work were funded by a Chan-Zuckerberg Initiative Essential Open Source Software grant to JGC; an Alfred P Sloan Foundation grant to JGC, CH, and JS; and the National Cancer Institute of the National Institutes of Health under the awards for the Partnership of Native American Cancer Prevention U54CA143924 (UACC) and U54CA143925 (NAU) to JGC. Initial development of the QIIME 2 platform was funded in part by the National Science Foundation grant 1565100 to JGC. Testing and initial application of this software was performed on Northern Arizona University's Monsoon computing cluster, funded by Arizona's Technology and Research Initiative Fund. Additional analysis effort was funded under the State of Arizona Technology and Research Initiative Fund (TRIF), administered by the Arizona Board of Regents, through Northern Arizona University.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

How to cite this article: Bolyen E, Dillon MR, Bokulich NA *et al.* **Reproducibly sampling SARS-CoV-2 genomes across time, geography, and viral diversity [version 2; peer review: 2 approved]** F1000Research 2020, **9**:657 https://doi.org/10.12688/f1000research.24751.2

First published: 29 Jun 2020, **9**:657 https://doi.org/10.12688/f1000research.24751.1

REVISED **Amendments from Version 1**

In this revision we have made minor changes to the text to address concerns of the reviewers or omissions that were pointed out by the reviewers, and added runtime and memory usage benchmark results. A new figure has been added to reflect this.

We have additionally released a new version of genome-sampler which expands the software documentation and simplifies the installation procedure.

Any further responses from the reviewers can be found at the end of the article

Introduction

The intersection of the SARS-CoV-2 outbreak and the genomics revolution has led to the rapid accumulation of viral genomes that are fueling our epidemiological understanding of the global pandemic. However, the rate of genome sequencing is challenging our ability to conduct comprehensive analyses in a timely manner. Local networks of health care professionals, laboratory professionals, and researchers are rapidly generating genome sequences at an unprecedented rate and feeding these data into global community resources, such as GISAID[1] and GenBank[2]. Contextualizing locally-derived genome sequences with those from global resources (e.g., as recently performed by the Arizona COVID-19 Genomics Union[3]) enables phylogenetic analyses that can provide information about the relative roles of local transmission versus repeated introductions. This can help to evaluate the utility of control measures, such as stay-at-home orders. These sequencing data thus enable a new paradigm in epidemiology, which must be facilitated by computational workflows designed to handle this scale of data.

Contextualization of locally derived genome sequences will generally begin with two collections of sequences: those obtained from a global community resource and those obtained locally. The widely used NextStrain[4] platform refers to these sequence collections in their documentation as the *context sequences* and the *focal sequences*, respectively, and we adopt that terminology here.

To enable phylogenetic analysis of full-length SARS-CoV-2 genomes, for example with Bayesian methods or maximum likelihood methods with bootstrap support, subsampling the context sequences is essential for computational feasibility. To avoid introducing post-sequencing sampling biases into our analysis, we subsampled the context sequences across three axes: time, space (i.e., geographic dispersion of near neighbors of focal sequences), and viral genome diversity. Sampling across time is a prerequisite to reliable inference of molecular clock signal from the data by ensuring that our sample of viral genomes span as much time as possible and include the oldest available genomes. Sampling the context sequences to include near neighbors of the focal sequences that come from different geographic regions enables us to avoid erroneously describing groups of focal sequences as monophyletic. Sampling across viral diversity enables us to represent the known diversity of the virus in our analysis. Each of these steps additionally reduces the chance of over-represented genomes dominating the analysis. When data sets are relatively small, this process can be performed manually, but when numbers of context genomes measure in the thousands, tens of thousands, or even hundreds of thousands (which may be likely as the pandemic progresses), an automated and reproducible subsampling approach is essential to maximize efficiency and to avoid human error.

Here we present `genome-sampler`[5], a QIIME 2 plugin that enables other research teams to apply our context sequence subsampling workflow. Our subsampling workflow is compatible with tools such as NextStrain[4], which includes a similar but not identical subsampling process (details provided in the *Discussion* section). We believe that our workflow can reduce sampling bias in analysis of SARS-CoV-2 genomes, and could be applied for regionally focused analyses, such as ours, or globally focused analyses. QIIME 2[6] (https://qiime2.org) is a plugin-based bioinformatics software platform developed for microbiome multi-omics analysis. It includes a unique retrospective data provenance tracking system that ensures reproducibility of bioinformatics steps by recording details of all analysis steps (commands called, parameters and input arguments provided, as well as details of the computational environment where the analysis was run, such as versions of underlying software dependencies; see examples at https://view.qiime2.org and in Figure 2 of the QIIME 2 paper[6]). We built this functionality as a QIIME 2 plugin because, given the pace at which SARS-CoV-2 genomics research is currently being carried out, human error in bioinformatics workflows is likely and the detailed record keeping needed to ensure reproducibility may be inadvertently skipped. QIIME 2 ensures that workflow errors could be detected retroactively and that workflows can be reproduced, even if detailed records are not kept while they are being run.

Methods

Implementation

`genome-sampler`[5] operates on three input files: a fasta file containing the unaligned context sequences, a fasta file containing the unaligned focal sequences, and a tab-separated text file containing metadata for the context sequences. The context sequences and metadata will typically be obtained by the user from a public repository such as GISAID. The focal sequences will typically be sequences that the team has compiled independently, for example from their locale.

Operation

`genome-sampler` can be installed in a conda environment on macOS or Linux systems, as described in its installation documentation linked from the project website. The complete workflow can be applied in one step using the included Snakemake[7] workflow, or the steps can be applied individually.

Most steps in `genome-sampler` run very quickly (within a few seconds to a few minutes), however two steps (`sample-diversity` and `sample-neighbors`) are much slower and highly dependent on dataset size and characteristics (Figure 1). A benchmark was performed on a single node of the monsoon cluster computer at Northern Arizona University with an Intel(R) Xeon(R) Gold 6132 CPU (28 logical processors)

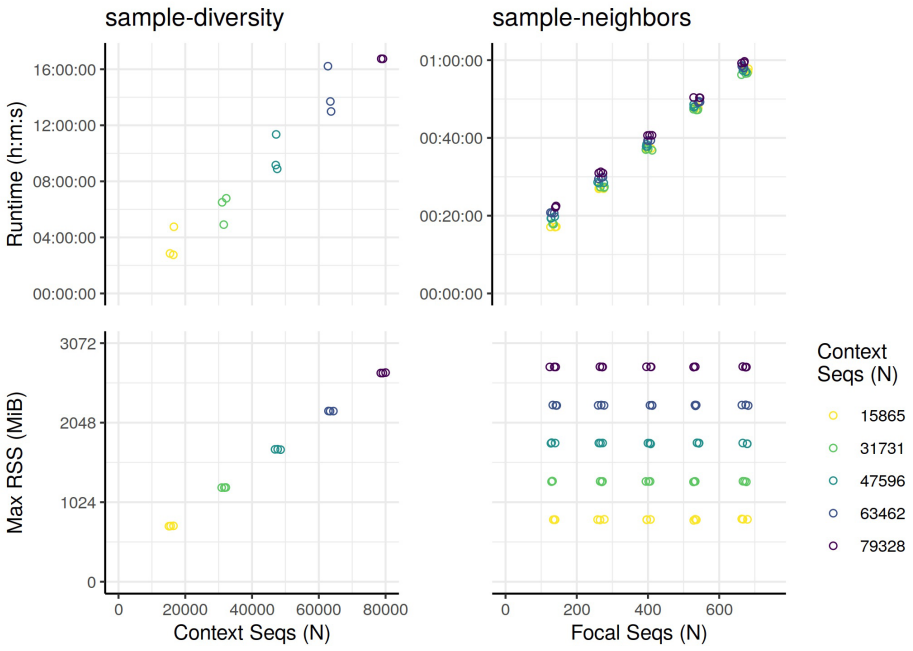


Figure 1. The runtime and memory requirements (top and bottom row, respectively) for `sample-diversity` and `sample-neighbors` (left and right column, respectively) are shown. Data was sourced from GISAID. Context sequences were resampled three times at each of the evenly spaced dataset sizes shown in the legend. The largest size (N=79,328) shows less variability because each subsample represented the entire dataset. The points are jittered on the x-axis to improve legibility, the y-axis remains un-jittered. *Max RSS* refers to Memory (max resident set size). All benchmarks were run with 28 threads (details in *Operation* section).

and 196 GB of RAM. Both `sample-neighbors` and `sample-diversity` were run with 28 threads. Context sequences were resampled randomly three times for each data-set size. The runtime of `sample-diversity` scaled with the number of sequences provided in a linear fashion, with variability related primarily to the specific characteristics of each initial random subsample. For `sample-neighbors`, the number of context sequences had a limited impact on the runtime, and was instead more directly related to the number of focal sequences. Memory (max resident set size) grew linearly with the number of context sequences for both steps.

Use case

Here we describe the series of steps taken by the `genome-sampler`[5] workflow (see Figure 2). In each step, any parameter values that can be overridden by the user are bolded. This description is accompanied by an online tutorial, available from the project website, which illustrates a use case focused on a small set of sequences obtained from GISAID. The tutorial is tested with each release of `genome-sampler` to ensure that all commands remain up to date.

The genome-sampler workflow works as follows:

1. Clean up and filter the context sequences.
   i. Filter sequences that contain non-IUPAC characters[8] as these characters can be problematic for downstream tools, such as sequence aligners or alignment viewers.
   ii. Remove any gap ("-" or ".") characters, as this workflow is intended to work on unaligned sequences. (Aligned reference sequences can be provided as input since they will be unaligned in this step.)
   iii. Filter sequences that are composed of **>10%** N characters.
   iv. Optionally filter sequences with length less than a user-specified minimum length or greater than a user-specified maximum length.
2. Uniformly sample context sequences across time, selecting **7** sequences from each **7**-day period between the **earliest** and latest dates represented in the data set. If there are fewer than 7 sequences in any 7-day period, all sequences from that period are included in the result. These sequences are referred to as the *temporally sampled context sequences*. The user can optionally supply a start date, in which case any genomes from before that time will be excluded.
3. Search focal sequences against context sequences to identify the **10** closest matches to each focal sequence. This is achieved using vsearch's `usearch_global` option[9] at **99.99** percent identity. The resulting collections of closest matches are sampled to select **3** geographically distinct context sequences for each focal sequence for inclusion in the subsampled context sequence collection. This sampling procedure is weighted such

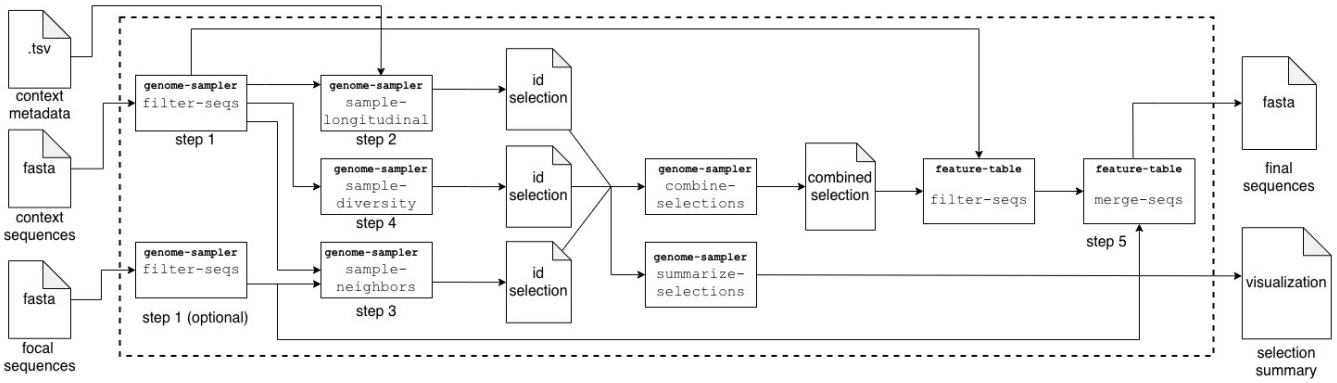


Figure 2. The `genome-sampler` workflow. This workflow samples context sequences for downstream phylogenetic analysis. Specific steps are represented by boxes: the QIIME 2 plugin name is bolded, and the action name in monospace font. Inputs and outputs are represented by folded-page file icons. The surrounding dashed box represents the Snakemake workflow which automates execution of the contained steps. Given context metadata, context sequences, and focal sequences, the Snakemake workflow will produce a fasta file which is ready for alignment and a summary of the sampling procedure as a QIIME 2 visualization.

that each geographic region has an equal probability of selection instead of each genome. This weighting prevents overrepresented regions from dominating the sample. This step ensures that any monophylies of target sequences are not artifacts of our sequence sampling approach. These sequences are referred to as the *geographically sampled context sequences*. (This step is achieved using sequence metadata, and can be parameterized so that this can be applied over any categorical metadata, not just geography.)

4. Cluster the complete context sequence collection with vsearch's `cluster_fast` option at **99.90** percent identity. The resulting cluster centroid sequences represent a divergent collection of the SARS-CoV-2 genomes and are referred to as the *diversity sampled context sequences*.

5. Combine the temporally, geographically, and diversity sampled context sequences with the focal sequence collection. The resulting collection of sequences will be deduplicated by sequence identifier, so sequences contained in multiple different subsamples are represented only once in the final sequence collection. This final collection of sequences should be used for downstream analysis.

Discussion

Resemblance to NextStrain context sequence sampling workflow

The NextStrain workflow also subsamples context sequences for its phylogenetic tree builds using augur (https://github.com/nextstrain/augur) and scripts in their ncov repository (https://github.com/nextstrain/ncov). Their workflow subsamples the context sequences across two axes: time and geography, prioritizing similarity to focal sequences when selecting sequences from different geographic regions. They sample across time by including a specified number of sequences per month for different regions. When determining the closest matches, percent identity is computed based on a multiple sequence alignment of all sequences, which is computed by aligning each sequence against a reference alignment using mafft[10].

Step 2 of our workflow is similar to their time sampling approach, but is independent of other variables such as geography. The workflows diverge more in Step 3, where we begin by identifying near neighbors of all focal sequences using global alignment search with vsearch. We then optionally sample across the geographic source of those sequences such that each geographic region represented in each collection of near neighbors has an equal probability of selection. We follow this with Step 4, where we sample the full genetic diversity of the context sequences by clustering them all against one another and including the resulting cluster centroid sequences in our final sequence collection. As far as we are aware, there is not an analog to our Step 4 in the NextStrain workflow.

Our workflow is modular in design to facilitate benchmarking and optimization of this essential context sequence sampling step. Our three sampling steps can be used individually or in any combination, and can be replaced with a random sampling step (the `sample-random` action) to allow evaluation of the importance of each step. At this stage, we do not claim that our workflow is better than the one used by NextStrain. We hope the similarity of our interfaces (both of which require the same input and output, are accessible through Snakemake, and use the same terminology to describe data) will allow for independent comparison of these and other approaches. In our next stage of work on this project, we plan to evaluate the impact of each subsampling step and their associated parameters on downstream phylogenetic results.

Retrospective data provenance tracking system

The retrospective data provenance tracking system implemented in QIIME 2 differs from other systems such as Snakemake[7] or Galaxy[11], which we view as providing prospective data provenance

tracking. For example, when a Snakemake file is used to run a workflow, that workflow is documented for reproducibility by the Snakemake file. However, if a user were to run the underlying commands independently, they must keep detailed records of their commands to ensure reproducibility of the analysis. This is not necessary with QIIME 2's retrospective data provenance tracking system, which records steps regardless of whether the workflow is run using a tool like Snakemake or Galaxy, or whether individual components are run independently. Additionally, QIIME 2's system assigns universally unique identifiers (UUIDs) to all execution steps, inputs, and outputs, so data can be unambiguously linked to workflow descriptions. QIIME 2 is therefore fully compatible with workflow engines such as Snakemake or Galaxy, but provides additional information which further ensures reproducibility.

We present `genome-sampler`[5], a QIIME 2 plugin that supports subsampling of genomic sequence collections based on time of genome isolation, geography of genome isolation, and genomic diversity, thus facilitating genomic epidemiology based on large numbers of genomes while reducing the possibility of post-sequencing sampling bias impacting conclusions. As the number of available SARS-CoV-2 genomes continues to increase rapidly, approaches such as this will be required to enable phylogenetic and other analyses of genome data.

Data availability

Source data

The context sequences and metadata used in the `genome-sampler` *Use case* were obtained from GISAID. Those genomes were sampled from patients in Arizona, USA, and published to GISAID by the Arizona COVID-19 Genomics Union (ACGU). The focal sequences and metadata used in the `genome-sampler` *Use case* were sequenced at a later time than the context sequences, also from patients in Arizona. The focal sequences were generated and assembled by the ACGU and are currently being added to GISAID. These context and focal sequences and associated metadata are all available for download for use in learning `genome-sampler` (see the project website). For analysis purposes, we recommend obtaining sequences from a public repository, such as GISAID or GenBank, as those sequences will be updated (for example to improve genome assemblies) before our tutorial data is updated.

Software availability

`genome-sampler` source code available at: at https://github.com/caporaso-lab/genome-sampler.

Archived source code and tutorial data at time of publication: https://doi.org/10.5281/zenodo.3891818[5].

License: BSD 3-Clause "New" or "Revised" License.

Documentation, written using Myst (https://myst-parser.readthedocs.io/en/latest/) and rendered using Jupyter Book (https://jupyterbook.org/), is available at http://caporasolab.us/genome-sampler/. If you need technical support, please post a question to the QIIME 2 Forum at https://forum.qiime2.org. We are very interested in contributions to `genome-sampler` from the community - please get in touch via the GitHub issue tracker or the QIIME 2 Forum if you're interested in contributing.

Acknowledgements

We are deeply indebted to the patients who provided SARS-CoV-2 samples (via an uncomfortable procedure) while suffering from this new pathogen, to the medical professionals who bravely collected the samples that form the basis for our genomic epidemiology work, and to the laboratory professionals who processed and sequenced genomes from these samples. Furthermore, we would like to acknowledge the GISAID sequence contributors from across the world who have sequenced genomes at an unprecedented rate and made them available for public use.

References

1. Elbe S, Buckland-Merrett G: **Data, disease and diplomacy: GISAID's innovative contribution to global health.** *Glob Chall.* 2017; **1**(1): 33–46.
PubMed Abstract | Publisher Full Text | Free Full Text

2. Benson DA, Cavanaugh M, Clark K, *et al.*: **GenBank.** *Nucleic Acids Res.* 2013; **41**(Database issue): D36–42.
PubMed Abstract | Publisher Full Text | Free Full Text

3. Ladner JT, Larsen BB, Bowers JR, *et al.*: **Defining the Pandemic at the State Level: Sequence-Based Epidemiology of the SARS-CoV-2 virus by the Arizona COVID-19 Genomics Union (ACGU).** *medRxiv.* 2020; 2020.05.08.20095935.
Publisher Full Text

4. Hadfield J, Megill C, Bell SM, *et al.*: **Nextstrain: real-time tracking of pathogen evolution.** *Bioinformatics.* 2018; **34**(23): 4121–4123.
PubMed Abstract | Publisher Full Text | Free Full Text

5. Center for Applied Microbiome Science, Pathogen and Microbiome Institute: **genome-sampler: Reproducibly Sampling SARS-CoV-2 Genomes Across Time, Geography, and Viral Diversity (Version 2020.6.0).** *Zenodo.* 2020.
http://www.doi.org/10.5281/zenodo.3891818

6. Bolyen E, Rideout JR, Dillon MR, *et al.*: **Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2.** *Nat Biotechnol.* 2019; **37**(8): 852–857.
PubMed Abstract | Publisher Full Text | Free Full Text

7. Köster J, Rahmann S: **Snakemake--a scalable bioinformatics workflow engine.** *Bioinformatics.* 2012; **28**(19): 2520–2522.
PubMed Abstract | Publisher Full Text

8. Cornish-Bowden A: **Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984.** *Nucleic Acids Res.* 1985; **13**(9): 3021–3030.
PubMed Abstract | Publisher Full Text | Free Full Text

9. Rognes T, Flouri T, Nichols B, *et al.*: **VSEARCH: a versatile open source tool for metagenomics.** *PeerJ.* 2016; **4**: e2584.
PubMed Abstract | Publisher Full Text | Free Full Text

10. Katoh K, Standley DM: **MAFFT multiple sequence alignment software version 7: improvements in performance and usability.** *Mol Biol Evol.* 2013; **30**(4): 772–780.
PubMed Abstract | Publisher Full Text | Free Full Text

11. Afgan E, Baker D, Batut B, *et al.*: **The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update.** *Nucleic Acids Res.* 2018; **46**(W1): W537–W544.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Peer Review Status: ✔ ✔

Version 2

Reviewer Report 18 January 2021

https://doi.org/10.5256/f1000research.30305.r73795

✔ **C. Titus Brown**
Department of Population Health and Reproduction, University of California, Davis, Davis, CA, USA

Thank you!

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: bioinformatics and software development

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 27 August 2020

https://doi.org/10.5256/f1000research.27305.r67936

? **C. Titus Brown**
Department of Population Health and Reproduction, University of California, Davis, Davis, CA, USA

- This paper describes a software package, genome-sampler, that subsamples collections of SARS-CoV-2 genomes with attention to various metadata attributes. The paper is well motivated and well written.

- My review focuses on the tutorial and getting the software running.
- I was pleased to see that the authors posted their source code to an archival location (Zenodo).
- I would encourage the authors to specify a release (rather than the latest branch) to install in the tutorial. Similarly, the conda install instructions in the tutorial should pin the versions of the software; this could easily be done via an environment yml provided within the genome-sampler github. And (minor nit) specifying '-y' in the copy/paste command line for conda install would be good too!
- The tutorial downloads FASTA data from a Dropbox. This is brittle and should be changed. I suggest using an archive (Zenodo, osf.io, DataDryad) for this data, and making it copy/paste downloadable. They're small enough to put in github and version, too.
- The tutorial downloads spreadsheet data from a Google Docs spreadsheet, and I would encourage the authors to put this data in git as well - even if there's a pedagogical reason to ask users to go through downloading from a spreadsheet, it's brittle to have a content-mutable and unversioned URL be the only location for the tutorial data.
- The approach of downloading the Snakefile is also brittle. I suggest revamping the tutorial so that it does a local install of a git clone; see e.g. this documentation (for an alpha package) that would be a simpler approach, I think:
  https://github.com/dib-lab/charcoal/tree/51caa9f034f3d301367cb6eea2ee96b5e1ea05bb#quickstart
- Note that snakemake supports config files that let you put the overridable configuration parameters in a YAML file. This might be nicer than editing the Snakefile directly. Happy to provide detailed examples on request.
- While I'm suggesting things, you could also use conda environments in the Snakefile to obviate the qiime etc. install. Then you'd just need to install snakemake and run it with --use-conda to get it all done.
- When running the tutorial, I see the following error:
  ...
  /qiime2/sdk/util.py", line 92, in parse_format
  raise TypeError("No format: %s" % format_str)
  TypeError: No format: GISAIDDNAFASTAFormat

  I'm not sure where to go from here. I've filed an issue with the full details.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: bioinformatics and software development

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Response 27 Aug 2020

C. Titus Brown, University of California, Davis, Davis, USA

I have managed to run the pipeline successfully, and have the following additional comments --

---

I successfully ran the tutorial, huzzah!

It would be good to add a rough estimate of resources required (memory, disk, CPU time) to the tutorial.

Competing Interests: No competing interests were disclosed.

Author Response 17 Oct 2020

Greg Caporaso, Northern Arizona University, USA

Thank you for the feedback on our manuscript, and for testing the software and offering suggestions. Below we provide a point-by-point reply. Your comments are presented in italics, and we reply to each in-line.

This paper describes a software package, genome-sampler, that subsamples collections of SARS-CoV-2 genomes with attention to various metadata attributes. The paper is well motivated and well written.

My review focuses on the tutorial and getting the software running.

I was pleased to see that the authors posted their source code to an archival location (Zenodo).

I would encourage the authors to specify a release (rather than the latest branch) to install in the tutorial. Similarly, the conda install instructions in the tutorial should pin the versions of the software; this could easily be done via an environment yml provided within the genome-sampler github. And (minor nit) specifying '-y' in the copy/paste command line for conda install would be good too!

We agree with the reviewer's suggestion and have updated our install instructions to install from a specific release. The conda installation now includes specific pinned versions of dependencies. (We prefer to not include the -y option, but rather have the user acknowledge that changes are about to be made to their system, so we have not included that.)

The tutorial downloads FASTA data from a Dropbox. This is brittle and should be changed. I suggest using an archive (Zenodo, osf.io, DataDryad) for this data, and making it copy/paste downloadable. They're small enough to put in github and version, too.

We also agree with this suggestion. As of our most recent release, the tutorial files are now packaged in the GitHub repository and are included in the updated Zenodo package that we have prepared for the most recent release and paper resubmission.

The tutorial downloads spreadsheet data from a Google Docs spreadsheet, and I would encourage the authors to put this data in git as well - even if there's a pedagogical reason to ask users to go through downloading from a spreadsheet, it's brittle to have a content-mutable and unversioned URL be the only location for the tutorial data.

As of our most recent release, these files are now packaged in the GitHub repository and are included in the updated Zenodo package.

The approach of downloading the Snakefile is also brittle. I suggest revamping the tutorial so that it does a local install of a git clone; see e.g. this documentation (for an alpha package) that would be a simpler approach, I think:
https://github.com/dib-lab/charcoal/tree/51caa9f034f3d301367cb6eea2ee96b5e1ea05bb#quickstart

We have updated the tutorial so that this file is downloaded from a release version of genome-sampler, which is less brittle. We prefer to not have users install from a git clone directly, since we are officially supporting conda installation (which tends to be easier for our novice user), and adding a second officially supported installation method would add to our technical support burden.

Note that snakemake supports config files that let you put the overridable configuration parameters in a YAML file. This might be nicer than editing the Snakefile directly. Happy to

provide detailed examples on request.

We thank the reviewer for this suggestion. We now use a Snakemake config file to configure these parameters.

While I'm suggesting things, you could also use conda environments in the Snakefile to obviate the qiime etc. install. Then you'd just need to install snakemake and run it with -- use-conda to get it all done.

This is an attractive option and we invested time in exploring this approach. We found that our platform-specific environment files complicate this, so we ultimately opted to not take this suggestion. Specifically, users would need to replace the environment path for every rule in the Snakefile (if they were using OS X instead of Linux, for example). We have simplified our installation workflow (both for Snakemake users and non-Snakemake users) and hope that our provided workflow can serve as a template for more sophisticated use-cases, such as rule-specific conda or singularity containers.

When running the tutorial, I see the following error:
...
/qiime2/sdk/util.py", line 92, in parse_format
raise TypeError("No format: %s" % format_str)
TypeError: No format: GISAIDDNAFASTAFormat
I'm not sure where to go from here. I've filed an issue with the full details.

We haven't come across this error, but it is likely related to genome-sampler not being installed correctly. Our updated install instructions should simplify this. We're happy to try to help work out the problem, but we haven't seen this issue on our issue tracker. We have not had reports of other users running into this (but we have had other support requests, so we know that this is not something that is preventing people from using genome-sampler).

Competing Interests: No competing interests were disclosed.

Reviewer Report 31 July 2020

https://doi.org/10.5256/f1000research.27305.r65756

James Hadfield
Vaccine and Infectious Disease Division, Fred Hutchinson Cancer Research Center, Seattle, WA, USA

This paper presents a software tool to tackle a pressing, but welcome, problem: the number of publicly shared SARS-CoV-2 sequences (c. 75,000 at the time of this review) are too numerous to be analysed or visualised using currently available methods in a time-frame relevant for understanding local outbreaks. There is a need for researchers to be able to interrogate a particular collection of samples in a wider (often worldwide) context, and the choice of such context will greatly influence the conclusions drawn. The approach presented here samples contextual sequences by considering temporal, geographical and sequence-similarity data.

As the paper notes, similar approaches are available and in use by the wider community, however there is value in creating a range of tools for researchers to employ as required and which best suit their needs. This tool is easily installable and provides a ready-to-use solution for a pressing problem. It is purposefully designed in such a way that it is interoperable with other approaches, and will be immediately useful to researchers.

I agree with the authors that a thorough evaluation and comparison of different subsampling approaches is beyond the scope of this paper, however there are some aspects which were not discussed in the manuscript which should be addressed (i.e. minor revisions required). Please note that this review focuses on the genome-sampler software described here, and is not a commentary on the wider QIIME 2 platform.

Installation

Following https://caporasolab.us/genome-sampler/tutorial.html, installation was straightforward and the provided example worked out of the box. The tutorial was well written and didn't require any background knowledge of QIIME 2. The authors should be commended for this.

Points to address

1. Time required. The example data provided (10 focal, 75 contextual sequences) took c. 40 minutes running on a laptop using a single core. As this tool will commonly be employed using all publicly available data as context (currently c. 75,000 genomes) an overview of the time (as well as memory & parallelizability) required to perform subsampling for various numbers of focal & contextual sequences should be provided.

2. Aligned genomes are not required as input as vsearch is used to compare genomes. My understanding is that vsearch will perform (a relatively fixed number of) pairwise alignments for each focal genome vs. the contextual data set to gauge percent identity. The paper would benefit from a short explanation of why this approach was used rather than aligning all sequences (e.g. to a reference genome).

3. There is no ability to subsample focal sequences. As the authors correctly mention in the introduction, the impressive rate of genome sequencing presents a number of challenges. It is already a reality that certain locally-derived (focal) datasets are large enough to require subsampling of their own, and this will become more common over time. The authors should address this by either implementing the ability to sample focal sequences or prescribing that the researcher must define a suitably small focal set.

4. The rapid submission of samples to public repositories should to be facilitated as much as possible. It may be commonplace to have focal samples which are also present in the

contextual data. Could the authors detail what would happen in this case (e.g. are the contextual "duplicates" removed or will they bias steps 3 & 4 as they may preclude the inclusion of other samples?), or does this use-case need to be avoided by the user?

Minor points

- Step 5 is not annotated on figure 1.
- [page 3, paragraph 3] Sampling across time won't necessarily allow the reliable inference of a clock signal, but is rather a prerequisite.
- [page 4, step 1(iii)] Are short sequences (i.e. those with lots of gaps) excluded here, or only those with a large proportion of Ns?
- [page 3, step 3] Reword to clarify that this step is performed per-sample, i.e. that the (10) closest matches are found for each sample in the focal set.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: I am involved in Nextstrain, which (as the paper mentions) provides a similar subsampling approach for SARS-CoV-2 sequences. (I am a proponent of seeing a diverse suite of tools developed which can be used interchangeably for genomic epidemiology research.)

Reviewer Expertise: Bioinformatics, phylogenetics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 16 Oct 2020

Greg Caporaso, Northern Arizona University, USA

Thanks for your thoughtful review of our manuscript! We have reviewed your comments

and are submitting a revision that addresses them as detailed here. The reviewer comments are presented in italics, and our replies to each follow in-line.

Installation

Following https://caporasolab.us/genome-sampler/tutorial.html, installation was straightforward and the provided example worked out of the box. The tutorial was well written and didn't require any background knowledge of QIIME 2. The authors should be commended for this.

Thank you!

Points to address

Time required. The example data provided (10 focal, 75 contextual sequences) took c. 40 minutes running on a laptop using a single core. As this tool will commonly be employed using all publicly available data as context (currently c. 75,000 genomes) an overview of the time (as well as memory & parallelizability) required to perform subsampling for various numbers of focal & contextual sequences should be provided.

In response to this request we have added a new figure to this paper presenting a benchmark of CPU time and memory usage for different sized context and focal genome data collections. Our results are presented in Figure 1.

Aligned genomes are not required as input as vsearch is used to compare genomes. My understanding is that vsearch will perform (a relatively fixed number of) pairwise alignments for each focal genome vs. the contextual data set to gauge percent identity. The paper would benefit from a short explanation of why this approach was used rather than aligning all sequences (e.g. to a reference genome).

In the ideal scenario, for our sample_diversity step, we would align all genomes in a collection against all other genomes in that collection, compute all similarities between genomes, and then use those pairwise similarities to cluster genomes based on their similarity. We could then select one or more sequences from each cluster for downstream analysis, which would ensure that we have represented the diversity of the genome collection. Computing all pairwise alignments is too time consuming for most practical applications however, so we use vsearch's heuristic approach which tries to achieve the same goal but by reducing the number of pairwise alignments that are computed by trying to prioritize alignments between each sequence and sequences that are suspected to be highly similar to it. This provides higher resolution than the all-against-one approach that the reviewer mentions, because for example, if two query sequences (say ACGTT and AGGTA) are aligned to a single reference sequence (say ACGTA), we know that both are 80% similar to the reference, but we don't know how similar those sequences are to each other (they could be 60% similar to each or 100% similar to each other in this example). Thus we could represent distance to a reference sequence reasonably well using the all-against-one comparison, but we wouldn't know if we had sampled the full diversity represented in the genome collection. Since we were interested in using the vsearch approach for the sample_diversity step, it made sense to us to also use this for the other steps in the

workflow.

There is no ability to subsample focal sequences. As the authors correctly mention in the introduction, the impressive rate of genome sequencing presents a number of challenges. It is already a reality that certain locally-derived (focal) datasets are large enough to require subsampling of their own, and this will become more common over time. The authors should address this by either implementing the ability to sample focal sequences or prescribing that the researcher must define a suitably small focal set.

It actually is possible to sample the focal sequences using the same approaches that are applied to the context sequences in our tutorial, though we acknowledge that that was not clear (and undocumented) in our original submission. In the revised documentation included with our updated release we have added a section that specifically discusses this point. See the new section, Sampling focal sequences.

The rapid submission of samples to public repositories should to be facilitated as much as possible. It may be commonplace to have focal samples which are also present in the contextual data. Could the authors detail what would happen in this case (e.g. are the contextual "duplicates" removed or will they bias steps 3 & 4 as they may preclude the inclusion of other samples?), or does this use-case need to be avoided by the user?

This is a great point which we did not address in our initial version of the documentation. This situation would bias the sample-neighbors step in particular, as those duplicated sequences would be very likely to be chosen as the nearest neighbors. We have added a note to the main tutorial, and a new section to the documentation Removing sequences present in both focal and context sequence collections, illustrating how the user can address this.

Minor points

Step 5 is not annotated on figure 1.

Thanks for pointing out this omission. We have updated the figure to fix this.

[page 3, paragraph 3] Sampling across time won't necessarily allow the reliable inference of a clock signal, but is rather a prerequisite.

We modified the text the reviewer is referring to clarify this point.

[page 4, step 1(iii)] Are short sequences (i.e. those with lots of gaps) excluded here, or only those with a large proportion of Ns?

Since the input sequences for genome-sampler are unaligned, any gap characters (- or .) are stripped from the sequences on import. At this step of the analysis, the user does have the ability to filter sequences based on their length (sequences with length less than a minimum length or greater than a maximum length, both of which can be specified by the user, are optionally filtered). This has been clarified in the text.

[page 3, step 3] Reword to clarify that this step is performed per-sample, i.e. that the (10) closest matches are found for each sample in the focal set.

We have modified the text to clarify this.

Competing Interests: No competing interests were disclosed.