sierra-local: A lightweight standalone application for secure HIV-1 drug resistance prediction

Jasper C Ho[1], Garway T Ng[1], Mathias Renaud[1], Art FY Poon[1,2]*

[1]Department of Pathology and Laboratory Medicine, Western University, London, ON, Canada; [2]Department of Microbiology and Immunology, Western University, London, ON, Canada.

*Corresponding author, apoon42@uwo.ca

Abstract

Genotypic resistance interpretation systems for the prediction and interpretation of HIV-1 antiretroviral resistance are an important part of the clinical management of HIV-1 infection. Current interpretation systems are generally hosted on remote webservers that enable clinical laboratories to generate resistance predictions easily and quickly from patient HIV-1 sequences encoding the primary targets of modern antiretroviral therapy. However they also potentially compromise a health provider's ethical, professional, and legal obligations to data security, patient information confidentiality, and data provenance. Furthermore, reliance on web-based algorithms makes the clinical management of HIV-1 dependent on a network connection. Here, we describe the development and validation of *sierra-local*, a local implementation of the Stanford HIVdb genotypic resistance interpretation system, which aims to resolve the ethical, legal, and infrastructure issues associated with remote computing. This open-source package reproduces the HIV-1 resistance scoring by the web-based Stanford HIVdb algorithm with a high degree of concordance (99.990%) and a higher level of performance than previous methods of accessing HIVdb programmatically.

INTRODUCTION

Genotype-based prediction of human immunodeficiency virus type 1 (HIV-1) drug resistance is an important component for the routine clinical management of HIV-1 infection [1, 2]. Detecting the presence of viruses carrying mutations that confer drug resistance enables physicians to select an optimal drug combination for that patient's treatment regimen. Furthermore, genotyping by bulk sequencing is a cost-effective alternative to the direct measurement of drug resistance from culturing virus isolates in a laboratory [3]. Provided access to affordable bulk sequencing at an accredited laboratory for clinical microbiology, the interpretation of HIV-1 sequences is the primary obstacle to utilizing resistance genotyping for HIV-1 care. Fortunately, there are several HIV-1 drug resistance interpretation algorithms – three of the most widely used being the Standard HIVdb [4], Agence Nationale de Recherche sur le SIDA (ANRS) AC11 [5], and Rega Institute [6] algorithms – that can be accessed at no cost through web applications or services hosted by remote

network servers. The Stanford HIVdb interpretation system can be accessed either through a web browser at http://hivdb.stanford.edu/hivdb or programmatically through its Sierra Web Service [7], which requires the transmission of an HIV-1 sequence from a local computer over the network to the remote server. This is a convenient arrangement for clinical laboratories because there is no need to install any specialized software, web browsers are ubiquitous and most users are familiar with submitting web forms.

On the other hand, there are a number of disadvantages to accessing interpretation systems over a network connection. First, HIV-1 sequences are highly sensitive patient information, not only because infection with HIV-1 remains a highly stigmatized condition, but also because sequence data have been used as evidence in the criminal prosecution of individuals for engaging in sexual intercourse without disclosing their infection status, leading to virus transmission [8]. Once sequence data have been transmitted to a remote server, one cedes all control over data security. Preventing the onward distribution of the data and deleting the data once the analysis is complete is entirely up to the administrators of the host server. Furthermore the unencrypted data are transmitted in the clear between a number of intermediary web servers, exposing these data to a 'man-in-the-middle' attack unless the host server employs a secure transfer protocol.

Second, the algorithm hosted on the server is effectively a black box — one has no insight into how resistance predictions are generated. Even if a version of the algorithm has been released into the public domain, one cannot be certain that the exact same algorithm was applied to their transmitted data. Importantly, the version of HIVdb used has been shown to affect resistance predictions generated, the general trend being an increase in both resistance scores and predicted resistance levels [9]. In addition to contributing to inconsistencies in algorithm outputs, this makes it difficult to track data provenance, *i.e.*, the historical record of data processing, that has become recognized as a critical gap in the workflows of clinical laboratories. For instance, the College of American Pathologists recently issued new accreditation requirements stipulating that clinical laboratories must track the specific version of software programs used to process patient data [10]. Thus, a reliance on web-based systems creates significant issues for the reproducibility and quality assurance of clinical workflows.

Third, dependence on a web resource may cause problems when the laboratory cannot access the host server, either due to local or regional network outages, or because the host server is mal-

functioning or offline. In our experience, the web servers hosting the more popular HIV drug resistance interpretation algorithms such as the Stanford HIVdb database are reliable and well-maintained. However, it is not unusual for other web-based algorithms developed and implemented by academic researchers to be relocated or go offline as developers move to other institutions or run out of resources or interest.

One of the important features of the Stanford HIVdb algorithm is that it is regularly updated and released into the public domain in a standardized XML-based interchange format — the Algorithm Specification Interface version 2 (ASI2) format [11] — that was formulated and published by the same developers in conjunction with the Frontier Science Foundation.

Here, we describe the implementation and validation of *sierra-local*, an open-source Python package for local execution of the HIVdb algorithm in the ASI2 format. This package utilizes, but does not require, a network connection to synchronize its local ASI2 file and reference data with the latest releases on the Stanford HIVdb web server. Our objective was to release a lightweight alternative to transmitting HIV-1 sequences to the HIVdb web server that minimizes the number of software dependencies, and that produces the exact same interpretations as the Sierra web service for all available HIV-1 sequences in the Stanford database.

MATERIALS AND METHODS

Data Collection. We obtained the entirety of the genotype-treatment correlation datasets available to us through HIVdb on May 7 2018. These data included nucleotide sequences of 105,694 protease (PR) isolates, 112,723 reverse-transcriptase (RT) isolates, and 12,332 integrase (IN) isolates for a total of 230,749 sequences. In addition to sequence data, each record comprised of a list of the specific antiretrovirals (ARVs) that each isolate had been exposed to *in vivo* prior to collection, the region and year of collection, and subtype as determined by the Stanford University HIV Drug Resistance Database's HIV Subtyping Program. After dividing the original dataset into gene-specific partitions of 1,000 records each and screening for empty and invalid data, the resulting dataset contained 103,711 PR entries, 111,222 RT entries, and 11,769 IN entries totalling 226,702 records.

Local HIVdb Algorithm Implementation. The team at Stanford HIVdb have created a Python tool, SierraPy (https://github.com/hivdb/sierra-client/tree/master/python), that serves as a command-line interface (CLI) for HIVdb. SierraPy does not process sequences directly, however, and only serves as a front-end for the HIVdb Sierra GraphQL Web Service (https://hivdb.stanford.edu/graphql) [7]. Its reliance on an active network connection to offload sequence processing to a remote server does not fulfill the usage gap we aim to address with *sierra-local*. To fill this gap, a system that provides a complete interface to a local version of the algorithm is needed. This local algorithm is first represented in its most raw form by the publicly available HIVdb ASI2 file, which encodes both the algorithm for resistance scoring sequences and annotations describing relevant drug resistance mutations (DRMs) and ARVs of interest [11]. In short, it serves as a container for the core of the HIVdb resistance interpretation system which is not directly usable in a data-processing pipeline, as it is essentially only a descriptive XML defining the rules by which sequences should be scored. In the existing HIVdb pipeline a Java-based interpretor generator called SableCC is used to compile an algorithm interpretor from the HIVdb ASI2 file, but there is currently no such compiler in Python that we were aware of. The usage of SableCC in our local implementation of the Stanford algorithm would introduce further dependencies and obfuscate the clear relationship between the algorithm file and how local interpretations are generated. Hence, in light of the need for a Python-based local interpretor for the ASI2 format, we developed a regular expressions-based keyword-parsing method by which *sierra-local* locally compiles an executable model in Python directly from the local algorithm file. This method iterates through the HIVdb algorithm as an XML tree object in Python 3 and extracts the information encoded within using ASI2 keywords defined by the ASI2 Document Type Definition (DTD). *sierra-local* then uses this method to create a model of dictionaries and lists that encapsulate drug clause definitions, drug class lists, resistance level interpretations, DRM comments, and complex drug-DRM scoring conditions. Once populated, this model serves as the framework for sequence resistance scoring.

Sequence Pre-Processing and Validation. Prior to scoring, the HIVdb Sierra Web Service performs several pre-processing and validation steps on submitted query sequences and identified mutation sites found within sequences. We emulated these steps as closely as possible to maintain fidelity to the HIVdb pipeline. Identified and emulated pre-processing steps include sequence

alignment, gene identification, mutation site classification, sequence trimming, sequence subtyping, and sequence validation. These pre-processing steps can be considered parts of the algorithm involved in generating resistance prediction scores that are not included and distributed in the HIVdb ASI2 file.

Sequence Alignment. Of particular importance in these steps is sequence alignment. HIVdb utilizes NucAmino [12] to initially align and identify amino acid mutations in each query sequence. This nucleotide-to-amino acid alignment program is optimized for viral gene sequences, and was developed by the HIVdb developers in the Go language. However, in practice, NucAmino does not return the aligned sequences themselves; instead it only returns a list of mutations relative to the consensus subtype B amino acid sequence and general sequence metadata, *e.g.,* the aligned start and end coordinates of the query sequence relative to HIV-1 *pol*. If *sierra-local* does not align the query viral sequences exactly as how HIVdb would, the mutations identified relative to the consensus subtype B sequence would not be identical in all cases. Since these mutations serve as the basis for resistance predictions, an identical alignment process is of the utmost importance in maintaining fidelity to HIVdb Sierra. Thus, the usage of NucAmino in our local implementation is practically mandated. We cannot substitute this alignment method for a more integrated, native approach in Python.

Therefore, *sierra-local* calls a pre-compiled NucAmino (version 0.1.3) binary as a Python subprocess with default settings to execute this pre-processing step. The resulting intermediate alignment information files are temporarily stored in the sequence file directory and parsed to obtain formatted mutations and sequence query descriptive metadata. Structuring our processing pipeline in this way means we cannot, from NucAmino, directly obtain the requisite level of information necessary to make our analyses and resistance predictions. Thus, we had to develop several extra methods in *sierra-local* to supplement the information found in the NucAmino results:

Gene Identification. A critical part of resistance scoring is knowing which gene of interest in HIV-1 *pol* is present in the sequence being analyzed. To do this, we use the HXB2 reference genome map of the nucleotide positions of the gene positions PR, RT, and IN within HIV-1 *pol*. With the aligned start and end positions of the query sequence parsed from the NucAmino results, we then identify the genes across which the sequence spans. In accordance with HIVdb, only the first gene

encountered is noted. That is, if multiple gene products of HIV-1 *pol* are encoded, PR is chosen first if possible, then RT, followed by IN.

Mutation Site Classification. In the process of mutation site classification, each amino acid mutation site identified by NucAmino was further categorized as an insertion, deletion, ambiguous, or mixture using the original nucleotide sequence as a reference for amino acid translation. For consistency, we modified and ported a method from the Sierra algorithm for determining nucleotide codon translation ambiguity from Java to Python. Each site of interest identified by NucAmino on the aligned and translated query sequence generated a list of characters representing possible encoded amino acids at that site. Associating mutation sites with more than one encoded amino acid allows for a 'fuzzy' matching of a single sequence to multiple scoring conditions sharing the same residue position but different mutations. In certain cases, 'highly ambiguous' sites encoding more than four possible amino acids – made possible due to the presence of ambiguous nucleotides such as 'R' (A/G), 'B' (C/G/T), and 'N' (A/T/C/G) – were flagged as 'ambiguous' and represented with a translation of 'X'.

Sequence Trimming. Leading and trailing regions in each sequence containing a minimum proportion (30%) of 'low quality' sites — defined as sequenced sites that are highly ambiguous, stop codons, unusual mutations, or frameshifts — with at least one of these sites every 15 residues were trimmed prior to resistance scoring. We define a site as sequenced if the codon does not have more than one unknown nucleotide. Sites further qualified as 'unusual mutations' if they were indicative of APOBEC-mediated G-to-A hypermutation [13] or if the highest frequency of that mutation in the pooled untreated and treated viruses for that specific group M subtype was less than 0.1% in the Stanford University HIV Drug Resistance Database. The reference data for APOBEC-mediated G-to-A hypermutations in HIV-1 *pol* and for other HIV-1 mutation prevalences are obtained from the Stanford HIVdb server and stored locally during the installation process of *sierra-local*, and are updated to accommodate changes in reference data over time.

Sequence Subtyping. The previously discussed trimming step requires that sequences be subtyped in order to determine the frequency of mutations in subtype-specific pooled untreated and treated viruses. We implemented a port of HIVdb's HIV Subtyping Program, which categorizes submitted sequences as a pure subtype, a circulating recombinant form (CRF), a unique recom-

binant form (URF), non-group M HIV-1, or HIV-2. This process calculates uncorrected pairwise distances (the proportion of nucleotide differences) between submitted nucleotide sequences and a set of 200 different subtype-specific reference sequences, after masking the World Health Organization's surveillance drug resistance mutations (SDRMs) found in the submitted sequence [14, 15]. Because the HIVdb HIV Subtyping Program is very simple and does not perform phylogenetic analysis or bootstrapping, its results may not be as accurate as more sophisticated systems more commonly used such as the Rega Institute HIV-1 Automated Subtyping Tool [4]. Comparing interpretation systems, it has been suggested that ANRS AC11, HIVdb, and Rega demonstrate discordances that may be subtype-dependent [16, 17]. Yet, current genotypic resistance interpretation systems, including Stanford HIVdb, are subtype-agnostic within themselves, meaning that they do not offer differential resistance penalty scoring based on the subtype identified. Thus, other than being used to trim submitted sequences, the subtyping in this system does not significantly influence HIV-1 sequence interpretation.

Sequence Validation. HIVdb Sierra also validates sequences and returns a list of validation problems found in each sequence. A sequence is invalidated or flagged as a query if: no genes are found; the sequence is a reverse complement; the genes have not been aligned properly; it is too short based on gene-specific minimum cutoff nucleotide lengths; it is trimmed at the 5' or 3' ends due to low-quality leading or trailing sites; an indel is longer than 30 base pairs; invalid 'NA' characters are found; one or more stop codons are found; too many 'unusual mutations' as previously described are found; the virus is subtyped as HIV-2; the number of APOBEC mutations is two or greater; the number of APOBEC mutations at drug resistance positions (DRPs) is one or greater; or if the count of frameshifts, unusual insertions, and unusual deletions together is positive. In short, there are many complex validation steps that need to be emulated in our system in order for *sierra-local* to provide the same information in all cases to the end-user.

Resistance Scoring. The HIVdb algorithm begins the scoring process by assessing each sequence's potential resistance to 22 commonly-used ARVs independently by searching for ARV-specific mutation conditions present in the query sequence. ARV-specific mutation conditions encode and define the circumstances in which a mutation in a particular gene has been shown to influence resistance, and quantifies this resistance with an integer penalty score. Each of these

mutation conditions are associated with one or more specified amino acid changes at one or more particular positions, the ARV of interest, and the gene of interest in HIV-1 *pol* targeted by the ARV. For a mutation condition's resistance score to be counted, all of these criteria must be fulfilled.

sierra-local iterates over the ARV-specific mutation conditions corresponding to the gene detected in the query sequence. Conditions in this set that are fulfilled—*i.e.* all required amino acid changes, singular or multiple, are present—are added to a temporary list as candidate conditions for resistance. In some small proportion of cases, candidate conditions may overlap completely with regards to HIV-1 *pol* sites. This overlap reflects multiple unique conditions at the same set of loci being fulfilled, stemming from nucleotide ambiguities present in the original query sequence. Once all candidates are generated, all overlapping candidates with sub-maximal scores are filtered out, ensuring that only the conditions with the highest resistance scores contribute to the total penalty score. This was the only method among multiple versions we implemented that correctly emulated HIVdb Sierra's choice of particular amino acids given a mixture mutation. Furthermore, this process allows for the accurate detection of multiple-site conditions comprised of individual DRMs that contribute to a positive resistance score as both single and combination DRMs.

Resistance Score Interpretation. Based on the resulting penalty scores, one of five different resistance level interpretations are given by *sierra-local* to each ARV assessed: susceptible, potential low-level resistance, low-level resistance, intermediate resistance, or high-level resistance. Each resistance level is associated with a certain numerical range of resistance scores, where a higher score indicates a higher level of predicted resistance. The HIVdb algorithm uses non-overlapping resistance score sum ranges for interpretation: susceptible: $(-\infty, 9]$; potential low-level resistance: $[10, 14]$; low-level resistance: $[15, 29]$; intermediate resistance: $[30, 59]$; high-level resistance: $[60, \infty)$ [7]. We use a simple linear search-based method that returns these descriptive levels based on the calculated sum of penalty scores.

Algorithm Output Generation. Once all queries are pre-processed, scored, and otherwise annotated, *sierra-local* writes these results into a JSON (JavaScript Object Notation) format that mimics the standard output format of the HIVdb Sierra Web Service.

For the sake of brevity and simplicity, we decided to have *sierra-local* omit the 'pretty pairwise' sequence output found in Sierra's standard output. This 'pretty pairwise' output usually contains a

numerical sequence of all residue positions, a reference amino acid sequence, an aligned nucleotide sequence, and a mutation-only list. These data are not critical to a rapid genotypic interpretation system and may be omitted without detriment to the central aim. Furthermore, their exclusion from the results leads to a five-fold reduction in the results file size, which may be significant when processing large batches of sequences.

We included results from the NucAmino alignment of each gene sequence: starting and ending aligned positions within the HIV-1 *pol* gene and a complete list of mutations found in the sequence. Each of these mutations is further annotated with the consensus amino acid, the position, the amino acid possibilities encoded by the translated codon, and whether the mutation is indicative of APOBEC-mediated G-to-A hypermutation [13], or caused by APOBEC (termed 'APOBEC DRM'), or an indel. Most importantly, in addition to gene sequence alignment results, *sierra-local* returns drug resistance interpretations in an identical format to Sierra. That is, for each submitted sequence, clinicians are provided with an overall score for each ARV, an inferred level of resistance based on the score, and a detailed breakdown of drug resistance mutations or conditions contributing to each total score. It is the inferred resistance to ARVs that primarily guide the clinical treatment of HIV-1 based on genotypic resistance predictions [18, 2]. Finally, *sierra-local* attaches individual validation warning results to each sequence entry. Query sequences may differ in validity and quality, and it is important for the end user to be aware of potential issues with the resistance predictions. After all, HIVdb's genotypic interpretation system is an algorithmic approach that by nature cannot fully capture the complexities of drug resistance and thereby relies on critical assumptions about the quality of the sequences.

These validation results take the form of a level and a message describing the cause of the warning. Potential levels take the form of three increasingly important levels: 'WARNING', 'SEVERE WARNING', and 'CRITICAL', each of which reflect different problems found in a sequence. For example, trimming of 'low-quality' sites or a sequence length below a certain threshold (80 nucleotides for PR; 200 nucleotides for RT; 200 nucleotides for IN) would result in a 'WARNING' validation flag. 'SEVERE WARNING' flags result from sequences having a length below an even lower threshold (60 nucleotides for PR; 50 nucleotides for RT; 100 nucleotides for IN). 'CRITICAL' sequences, which have no genes encoded, are not scored after validation.

Validation Against HIVdb Sierra. We scored the entirety of the partitioned and filtered HIVdb genotype–treatment correlation dataset as of May 7 2018 with both *sierra-local* and SierraPy, storing all output results files from both programs. Validation was conducted using the HIVdb version 8.5 algorithm on both platforms. Our primary outputs of interest were the predicted resistance resistance scores generated by our implementation, and their concordance to Sierra's predictions on the same data. Of secondary interest were mutation and sequence meta-data that accompany this data and which may help guide physician care: sequence alignment data; sequence subtype; validation results based on sequence length, quality, and completeness; DRM description comments and categorizations; and insertion/deletion (indel) positions, if present.

Subsequent analysis of validation testing was conducted in R (version 3.4.4) [19] using the *jsonlite* package (version 1.5) [20] to extract resistance scores and relevant meta-data from the JSON results files from either program. For each sequence and for each drug queried, total resistance scores, mutation-specific subscores, mutation comments and other meta-data were extracted from the SierraPy results files. These data were compared for identicality in a pairwise fashion with similarly extracted data from the corresponding *sierra-local* results files.

From this concordance analysis, we counted the sequences in each batch that had discrepancies between total and component resistance scores and resistance interpretations, and recorded these discordant sequence identifiers for further analysis and debugging. Furthermore, we identified the particular mutations contributing to discordance and manually identified recurrent patterns in these that reflected bugs in our software. We fine-tuned the development of the software using this information to accurately capture the predictive capabilities of Sierra in an iterative process.

Performance Measurements. To quantify the speed of *sierra-local* in processing sequences, we chose a random subsample of gene-specific batches balanced across the genes of HIV-1 *pol* in our filtered genotype-treatment correlation dataset. This sampled dataset comprised of 9,673 PR sequences, 10,000 RT sequences, and 9,720 IN sequences contained in 10 batches per gene. Each gene-specific batch, roughly 10,000 sequences in size, was independently processed with both *sierra-local* and SierraPy. This method was timed using the *time* package in Python to determine file run-times and the resulting processing speeds of each program, measured in sequences per second.

Program Interface. To make using the program easy for physicians used to the web interface of HIVdb, we also developed a lightweight graphical user interface (GUI). Coded in Python's native *Tkinter* GUI package, this interface does not require additional dependencies on the user-end and increases accessibility to the program across operating systems and platforms. Similar to the web interface of HIVdb's genotypic resistance prediction program, users may submit a FASTA file with an arbitrary number of sequences to analyze, or they may paste sequences into the program, making this process extremely accessible and lowering any technology barriers to interpreting their patients' HIV-1 viral sequences. For more advanced users that may use scripting or for those who prefer a more granular approach, a traditional command-line interface (CLI) is also available.

Software Availability. The source code for *sierra-local* may be found at http://www.github.com/PoonLab/sierra-local. *sierra-local* has also been released as a Python 3.x package accessible through the Python Package Index (https://pypi.org/project/sierralocal).

RESULTS

Implementing the HIVdb Algorithm from the Algorithm Specification Interface. Distribution of the Stanford HIVdb algorithm in an XML-based interchange format called the Algorithm Specification Interface version 2 (ASI2) enables a seamless approach to updating the algorithm and algorithm version control. Despite the accessibility of the HIVdb algorithm itself, the ASI2 file is necessary but not sufficient to generate resistance predictions. Additional steps needed but not encoded by the algorithm file include: sequence alignment, sequence quality control and validation, sequence trimming, sequence subtyping, and formatting of the results output file with resistance predictions and accessory meta-data. These processes comprise the bulk of the functionality developed in *sierra-local* and are coordinated to generate resistance predictions in a manner as identical as possible to HIVdb Sierra.

Fulfilling Data Provenance Obligations. As new versions of HIVdb are released to reflect the growing knowledge of HIV resistance, users may easily manually update their local copy of the algorithm with a provided Python script called `updater.py` or with an optional CLI flag. We also provide the option to automatically update the algorithm to the most recently released version, allowing users to generate resistance predictions according to current HIV-1 genotype-phenotype

correlations. An example of how this process prevents physicians and users from operating on outdated information is that during the development of this project, two major updates to the HIVdb algorithm were released to the public and changed how sequences would be scored. These changes were anticipated for and integrated into the *sierra-local* development process by updating the HIVdb ASI2 file each time an update was available. We anticipate further future updates to the algorithm from the team at Stanford and appreciate the algorithm release format.

The option to choose between automatic updating or version freezing of the algorithm enables physicians and researchers to fulfill potential software version tracking and data provenance requirements. However, even with functionalities addressing these obligations embedded in *sierra-local*, the health care provider still bears the responsibility of operating with a knowledge of their software and algorithm versions and the changes between these.

High Degree of Concordance with HIVdb Sierra. Out of the 103,711 PR, the 111,222 RT, and the 11,769 IN sequences (total 226,702) submitted to both *sierra-local* and HIVdb SierraPy, resistances scores and component subscores were completely identical in 226,680 (99.990%). Of the 22 sequences that did not have identical resistance scores for all ARVs between both programs, 5 were PR, and 17 were RT (Table 1). The continuing development of *sierra-local* will address these discordant cases as we implement successive validation steps present in the HIVdb processing pipeline. As well, improvements to the processing methods of *sierra-local* will ensure valid resistance score predictions according to the HIVdb algorithm for future sequences not yet published or contained within the genotype-treatment dataset used.

Increased Performance over HIVdb Sierra. Performance and hence, the speed, of software packages varies according to hardware, software, and input data characteristics. All development, testing, and validation was performed on a workstation running Ubuntu 18.04 LTS with an Intel Xeon E5-1650 v4 hexa-core CPU at 3.60 GHz and 16 GB of DDR4-2400 RAM with a gigabit network connection.

sierra-local achieved mean [range] processing speeds of 112.17 sequences/second (seq/s) [80.57, 124.15] (N=9673) for PR, 45.36 seq/s [36.30, 55.90] (N=10000) for RT, and 37.36 seq/s [36.44, 38.68] (N=9720) for IN. SierraPy, with the same dataset as previously described, yielded slower mean processing speeds of 16.01 seq/s [12.88, 17.60] for PR, 6.12 seq/s [4.83, 7.54] for RT, and

Attributed cause	PR	RT	IN
Differences in indel event handling	0	11	0
Differences in leading/trailing site trimming	3	5	0
Refusal to process due to no genes being found	2	1	0

Table 1 Discordant cases between *sierra-local* and SierraPy. Discordances between individual query results from *sierra-local* and SierraPy were manually inspected and primary causes of processing errors were identified. The majority of discordant cases occurred when the query sequence was of reverse-transcriptase. Overall, the few differences between resistance scores from *sierra-local* and SierraPy seem to result from unusual HIV-1 *pol* sequence features that may require additional pre-processing methods.

5.19 seq/s [5.05, 5.47] for IN. Although the size of sequence batches used in this performance comparison likely is a factor in the results by virtue of file writing and reading being done once per batch, the large batch size used minimizes the effect of these I/O processes on the overall runtime. Overall, *sierra-local* is able to process and return results for submitted query HIV-1 *pol* sequences approximately 7 times faster than SierraPy, depending on the nature of the sequences and the type of local computing resources available. This result is in the expected direction since local computing resources are able to be fully utilized, whereas SierraPy depends on server-side processing speed, server load, request balancing, as well as network speed and traffic. In this case, network speeds between SierraPy on the local workstation and HIVdb Sierra GraphQL Web Service were likely not a significant factor in the speed improvement results obtained. With slower network speeds and all other factors being equal, however, the relative processing speed of *sierra-local* to SierraPy can only be expected to increase.

DISCUSSION

The use of *sierra-local* enables genotypic resistance predictions generation in a secure environment independent of potential network outages which may be of particular importance to the clinical care of individual patients. Additionally, the performance benefits of *sierra-local* are evident in applications demanding larger numbers of sequences to analyze.

Processing Pipeline Differences from HIVdb Sierra. Despite the release of the source code of HIVdb Sierra partway through the development process of *sierra-local*, the emulation of a complete system that processes unaligned nucleotide sequences and produces identical resistance predictions and data summaries in a small and independent package was not a trivial task. We developed *sierra-local* with the aims of minimizing the number of additional programs users had to install and unifying the genotypic resistance interpretation system onto a single local machine. The only other possible system for a completely local implementation of the HIVdb algorithm is hosting an instance of the HIVdb Sierra Web Service itself, which was recently made possible with the release of the Sierra source code. This approach requires a sizeable number of software dependencies including: Apache Commons Lang, Apache Commons Math, Apache Commons IO, Apache Log4j, Google Guava, Google Gson, protonpack, and GraphQL-Java. Furthermore, hosting a stable web service for the sole purpose of independently generating clinical resistance predictions increases facility requirements for servers, information technology support staff, and generally complicates an already complex workflow. At times, choosing to circumnavigate the issue of independent web service hosting in favour of local algorithm interpretation limited our approach. These limitations manifested particularly while emulating functions not provided by NucAmino as previously described, but overall we believe this approach to be the most optimal for future development in base Python and for use in a clinical environment.

Ease of Use and Accessibility. Health providers may be used to the web interface of HIVdb which displays results in an interactive and responsive web page format. Hence, transitioning to a genotypic interpretation system that outputs results in a plain-text JSON format may present a challenge to some health care providers who have not used SierraPy previously. The format generated by either *sierra-local* or SierraPy is useful if further analysis is needed or if these programs are integrated into a data pipeline, but may not be intuitive to navigate or extract information from. For these reasons, we also developed an accessory module to *sierra-local* that generates local static web-pages from the initial JSON output of the program. This module is able to emulate the look and functionality of the user web interface for the Stanford HIVdb program given JSON results files from either *sierra-local* or SierraPy, further increasing the accessibility of the resistance information generated on a local machine. It would also provide an additional point of customization:

users may be able to select which information they want in their results, or even change the language the information is displayed in.

sierra-local supports health provider and patient access to information that is important to the clinical management of HIV-1 infections by addressing common technological, ethical, professional, and accessibility issues.

ACKNOWLEDGEMENTS

This work was supported in part by the Government of Canada through Genome Canada and the Ontario Genomics Institute (OGI-131) and by grants from the Canadian Institutes of Health Research (PJT-156178 and PJT-155990).

REFERENCES

[1] **Tural C, Ruiz L, Holtzer C, Schapiro J, Viciana P, González J, Domingo P, Boucher C, Rey-Joly C, Clotet B, et al.**. 2002. Clinical utility of HIV-1 genotyping and expert advice: the Havana trial. AIDS **16**:209–218.

[2] **Günthard HF, Aberg JA, Eron JJ, Hoy JF, Telenti A, Benson CA, Burger DM, Cahn P, Gallant JE, Glesby MJ, Reiss P, Saag MS, Thomas DL, Jacobsen DM, Volberding PA, International Antiviral Society-USA Panel**. July 2014. Antiretroviral Treatment of Adult HIV Infection: 2014 Recommendations of the International Antiviral Society-USA Panel. JAMA **312**:410–25. doi:10.1001/jama.2014.8722.

[3] **Mayer KH, Hanna GJ, D'Aquila RT**. March 2001. Clinical Use of Genotypic and Phenotypic Drug Resistance Testing to Monitor Antiretroviral Chemotherapy. Clin Infect Dis **32**:774–782. doi:10.1086/319231.

[4] **Liu TF, Shafer RW**. June 2006. Web Resources for HIV Type 1 Genotypic-Resistance Test Interpretation. Clinical Infectious Diseases **42**:1608–1618. doi:10.1086/503914.

[5] **Meynard JL, Vray M, Morand-Joubert L, Race E, Descamps D, Peytavin G, Matheron S, Lamotte C, Guiramand S, Costagliola D, et al.**. 2002. Phenotypic or genotypic resistance

testing for choosing antiretroviral therapy after treatment failure: a randomized trial. AIDS **16**:727–736.

[6] **Van Laethem K, De Luca A, Antinori A, Cingolani A, Perno CF, Vandamme AM**. 2002. A genotypic drug resistance interpretation algorithm that significantly predicts therapy response in HIV-1-infected patients. Antivir Ther **7**:123–129.

[7] **Tang MW, Liu TF, Shafer RW**. 2012. The HIVdb System for HIV-1 Genotypic Resistance Interpretation. Intervirology **55**:98–101. doi:10.1159/000331998.

[8] **Bernard EJ, Azad Y, Vandamme AM, Weait M, Geretti AM**. September 2007. HIV forensics: pitfalls and acceptable standards in the use of phylogenetic analysis as evidence in criminal investigations of HIV transmission. HIV Med **8**:382–387. doi:10.1111/j.1468-1293.2007.00486.x.

[9] **Hart SAS, Vardhanabhuti S, Strobino SA, Harrison LLJ**. June 2018. The Impact of Changes over Time in the Stanford University Genotypic Resistance Interpretation Algorithm. J Acquir Immune Defic Syndr **79**:1. doi:10.1097/QAI.0000000000001776.

[10] **Aziz N, Zhao Q, Bry L, Driscoll DK, Funke B, Gibson JS, Grody WW, Hegde MR, Hoeltge GA, Leonard DG, Merker JD, Nagarajan R, Palicki LA, Robetorye RS, Schrijver I, Weck KE, Voelkerding KV**. April 2015. College of American Pathologists' Laboratory Standards for Next-Generation Sequencing Clinical Tests. Arch Pathol Lab Med **139**:481–493. doi:10.5858/arpa.2014-0250-CP.

[11] **Betts BJ, Shafer RW**. June 2003. Algorithm Specification Interface for Human Immunodefiency Virus Type 1 Genotypic Interpretation. J Clin Microbiol **41**:2792–2794. doi:10.1128/JCM.41.6.2792-2794.2003.

[12] **Tzou PL, Huang X, Shafer RW**. 2017. Nucamino: a nucleotide to amino acid alignment optimized for virus gene sequences. BMC Bioinformatics **18**:138. doi:10.1186/s12859-017-1555-6.

[13] **Mangeat B, Turelli P, Caron G, Friedli M, Perrin L, Trono D**. July 2003. Broad antiretroviral defence by human APOBEC3G through lethal editing of nascent reverse transcripts. Nature **424**:99–103. doi:10.1038/nature01709.

[14] **Bennett DE, Camacho RJ, Otelea D, Kuritzkes DR, Fleury H, Kiuchi M, Heneine W, Kantor R, Jordan MR, Schapiro JM, Vandamme AM, Sandstrom P, Boucher CA, van de Vijver D, Rhee SY, Liu TF, Pillay D, Shafer RW**. March 2009. Drug Resistance Mutations for Surveillance of Transmitted HIV-1 Drug-Resistance: 2009 Update. PLoS One **4**:e4724. doi:10.1371/journal.pone.0004724.

[15] **Wensing AM, Calvez V, Gunthard HF, Johnson VA, Paredes R, Pillay D, Shafer RW, Richman DD**. 2014. 2014 Update of the Drug Resistance Mutations in HIV-1. Top Antivir Med **22**:642–650. doi:10.1097/QAI.0000000000000072.

[16] **Snoeck J, Kantor R, Shafer RW, Van Laethem K, Deforche K, Carvalho AP, Wynhoven B, Soares MA, Cane P, Clarke J, Pillay C, Sirivichayakul S, Ariyoshi K, Holguin A, Rudich H, Rodrigues R, Bouzas MB, Brun-Vezinet F, Reid C, Cahn P, Brigido LF, Grossman Z, Soriano V, Sugiura W, Phanuphak P, Morris L, Weber J, Pillay D, Tanuri A, Harrigan RP, Camacho R, Schapiro JM, Katzenstein D, Vandamme AM**. February 2006. Discordances between Interpretation Algorithms for Genotypic Resistance to Protease and Reverse Transcriptase Inhibitors of Human Immunodeficiency Virus Are Subtype Dependent. Antimicrob Agents Chemother **50**:694–701. doi:10.1128/AAC.50.2.694-701.2006.

[17] **Wagner S, Kurz M, Klimkait T**. 2015. Algorithm evolution for drug resistance prediction: comparison of systems for HIV-1 genotyping. Antivir Ther **20**:661–665. doi:10.3851/IMP2947.

[18] **Paredes R, Tzou PL, Van Zyl G, Barrow G, Camacho R, Carmona S, Grant PM, Gupta RK, Hamers RL, Harrigan PR, Jordan MR, Kantor R, Katzenstein DA, Kuritzkes DR, Maldarelli F, Otelea D, Wallis CL, Schapiro JM, Shafer RW**. July 2017. Collaborative update of a rule-based expert system for HIV-1 genotypic resistance test interpretation. PLoS One **12**:e0181357. doi:10.1371/journal.pone.0181357.

[19] **R Core Team**. 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria.

[20] **Ooms J**. 2014. The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. arXiv:1403.2805 [stat.CO] .