Kive: a framework for version control of bioinformatics pipelines and data, and its application to HIV resistance genotyping

Richard H Liang[1], Eric Martin[1], Rosemary M McCloskey[1], Don Kirkby[1], James Nakagawa[1], Joshua Horacsek[1], T. Nguyen[1], Hope Lapointe[1], Chanson J Brumme[1], P Richard Harrigan[1,2], and Art FY Poon[1,2,3]

IAS 2015 vancouver, canada

TUPEA059

[1]BC Centre for Excellence in HIV/AIDS, Vancouver, Canada;
[2]Department of Medicine, University of British Columbia, Vancouver, Canada
[3]Faculty of Health Sciences, Simon Fraser University, Burnaby, Canada

CONTACT: Art Poon
BC Centre for Excellence in HIV/AIDS
680-1081 Burrard St.
Vancouver, BC V6Z 1Y6

apoon@cfenet.ubc.ca

BACKGROUND

- Bioinformatic "pipelines" are collections of software programs that are used to process and analyze biological data.
- Pipelines have become essential tools in modern biomedical and clinical laboratories.
- Most pipelines are customized to meet the requirements of each lab and project. Therefore they are usually under constant development.
- The end-users are often unaware of revisions being made to pipelines.
- It can be difficult to determine which version of a pipeline was used to process a given data set, especially when there are multiple copies of results.
- This makes it difficult to reproduce results for method validation or publication.
- Clinical laboratory accreditation programs (such as the College of American Pathologists, CAP) have issued new requirements for the validation and version tracking of bioinformatic pipelines.
- A system for tracking this information should make it possible to look up the pipeline history of any data set. It should be easy to use, with an intuitive graphical interface, and with as much of the "bookkeeping" automated as possible. We could not find a system that met these criteria.

OBJECTIVE

- To develop a new accessible computing framework for the version control of bioinformatic pipelines and their products.
- To use this framework in the development and validation of a pipeline for HIV resistance genotyping by next-generation sequencing.

METHODS

We developed our new framework ("Kive") as a Django application. Django is a Python framework for developing web applications.

Kive is built on a PostgreSQL relational database. The database records the digital "fingerprint" (md5 checksum) of every version of pipeline components and data sets, their locations in the filesystem, and their relations to each other.

Executing a pipeline version on a data set is completely automated by Kive, which distributes jobs across computing resources (such as a computing cluster) and records every intermediate step in the database. Any intermediate step that can be re-used in subsequent pipeline versions will be loaded to minimize computing time.

Read/write privileges to pipelines and data sets in Kive are specific to users and groups.

Kive also features a web-based graphical user interface, including a point-and-click toolkit for assembling and running pipelines that is implemented in HTML5 Canvas and JavaScript.

We used Kive to track versions of pipelines being developed in-house for processing and interpreting raw data sets from an Illumina MiSeq. This pipeline comprises 8 scripts written in Python, Ruby, and R.

EXAMPLE: APPLICATION TO BAD CYCLES

Figure 1. "Bad cycles" in a MiSeq run.
This plot summarizes the empirical tile- and cycle-specific MiSeq error rates based on reads covering the ΦX174 control template. In this run targeting small HIV RT amplicons, there were unusually high error rates (labelled directly on the forward-reads plot below) affecting a small number of tile-cycle combinations. Bases corresponding to these bad cycles were not flagged by Illumina's MiSeqReporter software with low quality scores.

These bad cycles will systematically affect all amplicon-based samples in a run. One cycle in particular caused significant overestimates in the frequencies of E138A (which decreases susceptibility to some NNRTIs) in the samples being processed in this run for resistance genotyping.

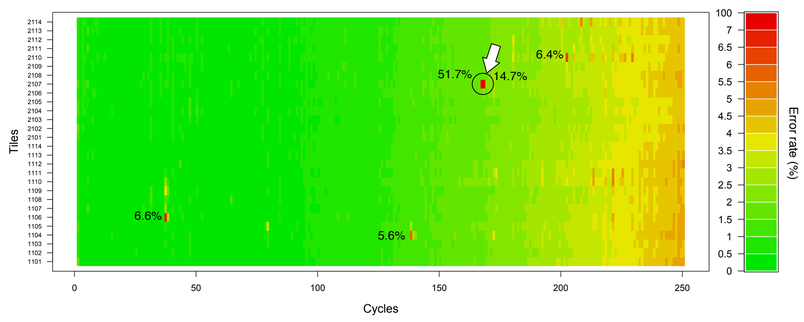


Figure 2. Revision to the MiSeq pipeline in Kive.
In response to the observation of "bad cycles" in MiSeq control data, we developed a filtering method that discards these cycles on the basis of empirical error rates as determined from reads covering the ΦX174 control template.

These screen captures illustrate how pipeline versions are presented to the user in Kive. These diagrams correspond to MiSeq pipeline versions 6.6 (**top**) and 6.7.3 (**bottom**, with error rate filtering), respectively. Grey shapes represent "methods" — bioinformatic scripts that comprise the pipeline. Green cylinders represent raw data inputs. Blue cubes indicate "structured" data inputs (data written in a predefined CSV format with specific variable types). Red cylinders represent the pipeline data outputs.

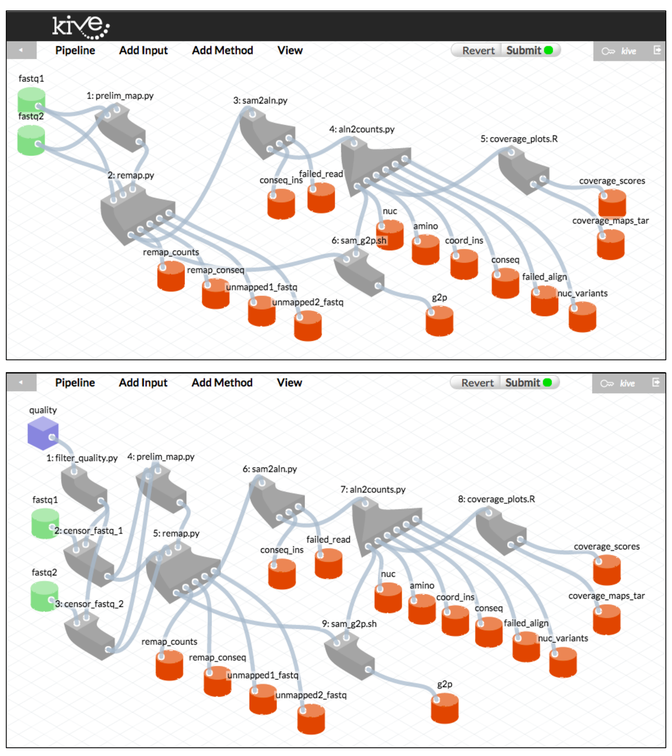


Figure 3. Retrieving a pipeline history in Kive.
When the user "drags" a pipeline output file into the web browser window, Kive displays (see screen capture below) the raw inputs and the exact version of the pipeline that produced this output (highlighted in blue) based on that file's md5 checksum. This history is permanently recorded in the Kive database.

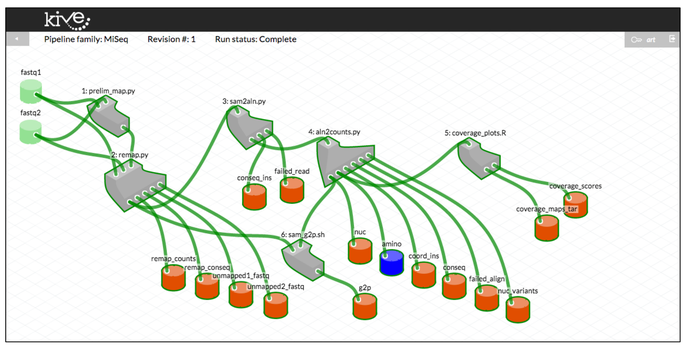
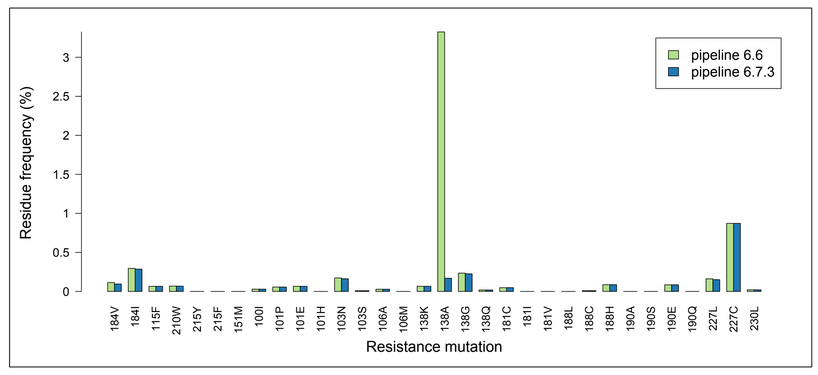


Figure 4. Outcome of pipeline revision.
This barplot depicts the observed frequencies of amino acids that are associated with resistance to NRTIs or NNRTIs within HIV RT codons 90-234 (HXB2 coordinates; drug resistance associations according to Stanford HIV Database). These frequencies were generated for a single pNL4-3 clonal sample processed with pipeline versions 6.6 and 6.7.3. This clone has HXB2 "wildtype" residues at all listed positions. Thus, any minority variant frequency should reflect experimental or bioinformatic error.

We observed >3% prevalence of E138A in this particular sample for pipeline version 6.6. However, applying the empirical error rate filtering in version 6.7.3 substantially reduced this frequency.



DISCUSSION

- Kive is being used to track the development and testing of a core pipeline for Illumina MiSeq data at the BC Centre for Excellence in HIV/AIDS Laboratory Program.
- This framework greatly facilitates the laboratory in meeting new CAP requirements for tracking bioinformatic processing.
- Kive provides a transparent environment for the exchange of bioinformatic resources between "developer" and "end-user" members of a laboratory.
- It becomes easy to recover the exact methods used to generate a specific set of results, months (or years) after the analysis — for instance, when submitting work for peer review.
- Kive is designed to utilize a clustered computing environment if one is available, and to reuse intermediate data produced by steps shared between pipeline versions.

AVAILABILITY

kive

Kive is an open-source project (all programs are freely available for download, use, and modification).
It is released under the BSD free software license and can be obtained at http://github.com/cfe-lab/Kive.

This work was supported by grants from the Canadian Institutes for Health Research (CIHR HOP-111406 to AFYP) and the Genome Canada-CIHR Partnership in Genomics and Personalized Health (Large-Scale Applied Research Project, PRH). AFYP is supported by a CIHR New Investigator Award and by a Career Investigator Scholar Award from a partnership between the Michael Smith Foundation for Health Research, St. Paul's Hospital Foundation, and the Providence Health Care Research Institute. PRH is supported by a CIHR/GlaxoSmithKline Research Chair in Clinical Virology.

GenomeBritishColumbia GenomeCanada Providence HEALTH CARE How you want to be treated. a place of mind UBC