Bioinformatics, 32(8), 2016, 1275–1277
doi: 10.1093/bioinformatics/btv724
Advance Access Publication Date: 12 December 2015
Applications Note

OXFORD

Databases and ontologies

Sequence database versioning for command line and Galaxy bioinformatics servers

Damion M. Dooley[1,*], Aaron J. Petkau[2], Gary Van Domselaar[2] and William W.L. Hsiao[1,3,*]

[1]Department of Pathology, University of British Columbia, Vancouver, BC, Canada, [2]National Microbiology Laboratory, Public Health Agency of Canada, Winnipeg, MB, Canada and [3]BC Public Health Microbiology and Reference Laboratory, Vancouver, BC, Canada

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on 12 August 2015; revised on 1 December 2015; accepted on 6 December 2015

Abstract

Motivation: There are various reasons for rerunning bioinformatics tools and pipelines on sequencing data, including reproducing a past result, validation of a new tool or workflow using a known dataset, or tracking the impact of database changes. For identical results to be achieved, regularly updated reference sequence databases must be versioned and archived. Database administrators have tried to fill the requirements by supplying users with one-off versions of databases, but these are time consuming to set up and are inconsistent across resources. Disk storage and data backup performance has also discouraged maintaining multiple versions of databases since databases such as NCBI nr can consume 50 Gb or more disk space per version, with growth rates that parallel Moore's law.

Results: Our end-to-end solution combines our own Kipper software package—a simple key-value large file versioning system—with BioMAJ (software for downloading sequence databases), and Galaxy (a web-based bioinformatics data processing platform). Available versions of databases can be recalled and used by command-line and Galaxy users. The Kipper data store format makes publishing curated FASTA databases convenient since in most cases it can store a range of versions into a file marginally larger than the size of the latest version.

Availability and implementation: Kipper v1.0.0 and the Galaxy Versioned Data tool are written in Python and released as free and open source software available at https://github.com/Public-Health-Bioinformatics/kipper and https://github.com/Public-Health-Bioinformatics/versioned_data, respectively; detailed setup instructions can be found at https://github.com/Public-Health-Bioinformatics/versioned_data/blob/master/doc/setup.md

Contact: Damion.Dooley@Bccdc.Ca or William.Hsiao@Bccdc.Ca

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

As outlined in motivation, the challenge is to efficiently archive versions of large FASTA format reference sequence databases which usually grow with many inserts but relatively few deletes or updates. These databases are suited to differential archiving (hereafter referred to as 'diff') in which the differences between consecutive version files are encoded as a set of instructions that enables their regeneration. With this in mind, we reviewed existing solutions for large text file versioning using the following criteria:

- A choice of archiving engines tuned to file content.
- An efficient differential versioning engine for large (>30 Gb) key-value text files.

- Handle scheduled import of reference databases.
- Easy interface to Galaxy.

We concluded that a large FASTA file differential archiving system would have to be developed. Some NCBI reference databases provide daily diff files spanning the most recent month (e.g. ftp://ftp.ncbi.nih.gov/genbank/daily-nc/) but no publically available client-side system exists for version update and retrieval. Git (http://git-scm.com), a file and code versioning tool, was evaluated as a possible solution but was found to be inefficient at versioning large FASTA files. As noted by Ram (2013) Git does not handle large datasets well, and must externalize them as separate files using tools like git-annex (https://git-annex.branchable.com). Our review of popular key-value databases (see Supplementary Data) found that they lack a versioning system for key-value contents.

2 Implementation

Our end-to-end versioned Data System combines BioMAJ version 1.2.3 (Filangi *et al.*, 2008) a flexible reference database download manager, with several versioning tool options and a user-friendly graphical interface (Galaxy). Figure 1 shows our proof-of-concept implementation of a Galaxy interface to the Kipper, git, file-folder and Biomaj systems.

BioMAJ places large multi-volume reference databases into a versioned folder structure, and can trigger download post-processes on them like Kipper diff transformation. Recently the BioMAJ team introduced a Galaxy tool that connects current BioMAJ downloads (https://www.e-biogenouest.org/resources/1397/download/AnthonyBretaudeau_Galaxy_Day_IFB_2014_BioMAJ2Galaxy.pdf). This nicely addresses the end-to-end data synchronization problem for current databases, but database versions are stored in full, so the storage space problem remains, and recall functionality can still be enhanced.

2.1 Kipper

Kipper is a command-line file versioning solution we have created for key-value text records. It manages a simple key-value data store by keeping track of whether a given key is inserted, updated, or deleted for each version. It recalls versions by date or version id, and stores them in separate volume files when convenient. An additional JSON metadata file catalogs the data store's name, data type, volume(s) and versions. Version metadata can be downloaded separately to determine if a new version is available. Kipper currently accepts text files having tab or space delimited key-value records, as well as the standard multi-line FASTA format. For FASTA data, the unique sequence identifier (text between '>'and first space) is used for key text. The remaining FASTA description and sequence becomes the value.

In a volume each key-value record is stored as a line containing the following tab-delimited fields (see Table 1): creation version id (Ins), deletion version id (Del), key (Key) and value (Value). As they are sorted by key and then creation version id, consecutive records track a key-value's inserts, updates and deletes. Table 1 shows an update to the description text for a FASTA RCSB Protein Database record with gene id 384482242. The first line shows the original record was inserted in (our) version 1. It is removed in version 3, and the following line re-inserts it with an additional associated gene id in version 3. The sequence is removed under this key in version 7 (elsewhere it actually lives on as a new gi|817598624|pdb|4D0C|C record).

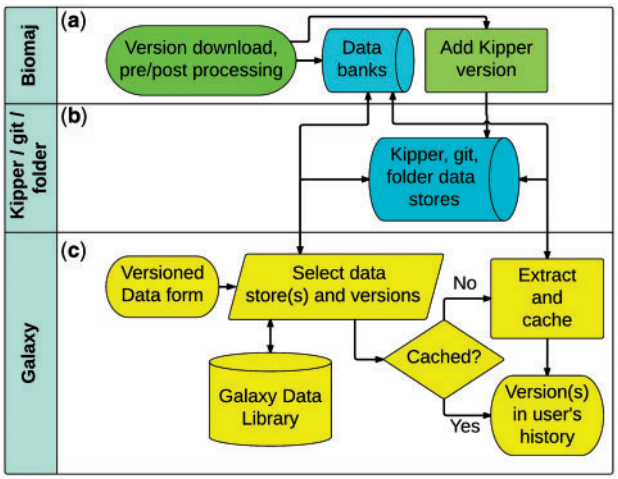

Fig. 1. End-to-end database versioning system: (**a**) BioMAJ on a schedule checks for and downloads new database versions. For a given database it can then trigger Kipper to add it to a volume. (**b**) One can also add new database versions directly via command line either to a Kipper or git archive or as a new sub-folder. (**c**) Galaxy can then provide these versions to users for retrieval

Table 1. Kipper record format example

Ins	Del	Key	Value
1	3	gi|384482242| pdb|2YF5|C	Chain C, Complex Of A B21 Chicken Mhc Class I Molecule And A 10mer Chicken Peptide TAGQSNYDRL
3	7	gi|384482242| pdb|2YF5|C	Chain C, Complex Of A B21 Chicken Mhc Class I Molecule And A 10mer Chicken Peptide**^Agi|817598624| pdb|4D0C|C Chain C, Complex Of A B21 Chicken Mhc Class I Molecule And A 10mer Chicken Peptide** TAGQSNYDRL

2.2 Galaxy versioned data tool

Like many labs, we are adopting Galaxy (Giardine *et al.*, 2005) for running bioinformatics tasks and workflows. Within this context, our Versioned Data tool provides an easy interface for retrieving multiple reference databases and derived products like NCBI-BLAST databases. This tool enables a user to select versioned datasets by name from a list (Fig. 1c). In a special Galaxy data library called 'Versioned Data' a Galaxy admin can arrange versioned data first by data source (NCBI, EBI, etc.) or by type (viral, bacterial, etc.) Individual data stores are set up within this hierarchy and are listed in the tool. A user can then select the current version, or a particular retrieval date or version number. Requested datasets are fetched from a cache if they exist there, otherwise the cache is populated with newly extracted data. Folder and BioMAJ interfaces do not need caching—they merely return links to permanent folder content.

A Galaxy admin can also set up a set of post-processing workflows, such as BLAST (Camacho *et al.*, 2008) indexing, that users can trigger directly on retrieved datasets. The results are cached for reuse by other users. A cache-cleaning script can be run periodically to remove all but the latest cached version of any dataset.

3 Discussion

We have found that Kipper can version 50 Gb+ text files that other archiving systems like git have no capacity for. There are a few

operational issues we plan to address as detailed in the Supplementary Data file (e.g. small changes in FASTA descriptions can lead to re-insertions of long sequences; also the prerequisite download of large input files can be unreliable.)

The standardization of reference databases by date of publication is a key problem for experimental replication in the bioinformatics realm. The community needs to develop a basic metadata standard for reference database sharing. Towards this goal, our Kipper versioning and archiving system brings reference database reproducibility, ease of use and lower maintenance costs to existing server infrastructure. We welcome partnerships to extend the Kipper data store functionality, and we encourage reference database providers to consider Kipper for convenient version storage, recall and publishing.

Acknowledgements

More information about the project can be found at http://www.irida.ca.

Funding

This work was supported by the Genome Canada/Genome BC Grant 172PHM 'A Federated Bioinformatics Platform for Public Health Microbial Genomics' under Fiona Brinkman, Gary Van Domselaar and William Hsiao.

Conflict of Interest: none declared.

References

Camacho,C. *et al*. (2008) BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.

Filangi,O. *et al*. (2008) BioMAJ: a flexible framework for databanks synchronization and processing. *Bioinformatics*, **24**, 1823–1825.

Giardine,B. *et al*. (2005) Galaxy: a platform for interactive large-scale genome analysis. *Genome Res*., **15**, 1451–1455.

Ram,K. (2013) Git can facilitate greater reproducibility and increased transparency in science. *Source Code Biol. Med*., **8**, 7.