CROSS-LINGUISTIC DATA ... IS CODE

Robert Forkel

April 30, 2015

Max Planck Institute for evolutionary Anthropology

Code is data

Homoiconicity in programming languages means:

> *code can be treated as a basic data structure that the programming language knows how to access.* -
>
> `http://blogs.mulesoft.org/code-is-data-data-is-code/`

Thus the programming language can manipulate code more reliably (no syntax errors!) than your text editor.

So homoiconicity is a desirable property of a programming language, because it allows for better tooling.

Data is code

Turns out – in a rather mundane way – data is code, too.

Or less catchy: cross-linguistic data is similar enough to (open source) code to share its tools:

- textual data

Data is code

Turns out – in a rather mundane way – data is code, too.

Or less catchy: cross-linguistic data is similar enough to (open source) code to share its tools:

- textual data
- line based (thanks to Harald and cldf)

Data is code

Turns out – in a rather mundane way – data is code, too.

Or less catchy: cross-linguistic data is similar enough to (open source) code to share its tools:

- textual data
- line based (thanks to Harald and cldf)
- not really Big Data

Data is code

Turns out – in a rather mundane way – data is code, too.

Or less catchy: cross-linguistic data is similar enough to (open source) code to share its tools:

- textual data
- line based (thanks to Harald and cldf)
- not really Big Data
- open (often)

DATA CURATION WITH GITHUB

Overview

Using

- **cldf** as format for our data,
- **git** (a tool for distributed source code management) and
- **GitHub** (a platform hosting git repositories).

we get a platform for collaboratively curating cross-linguistic data.

Example: Tsammalex

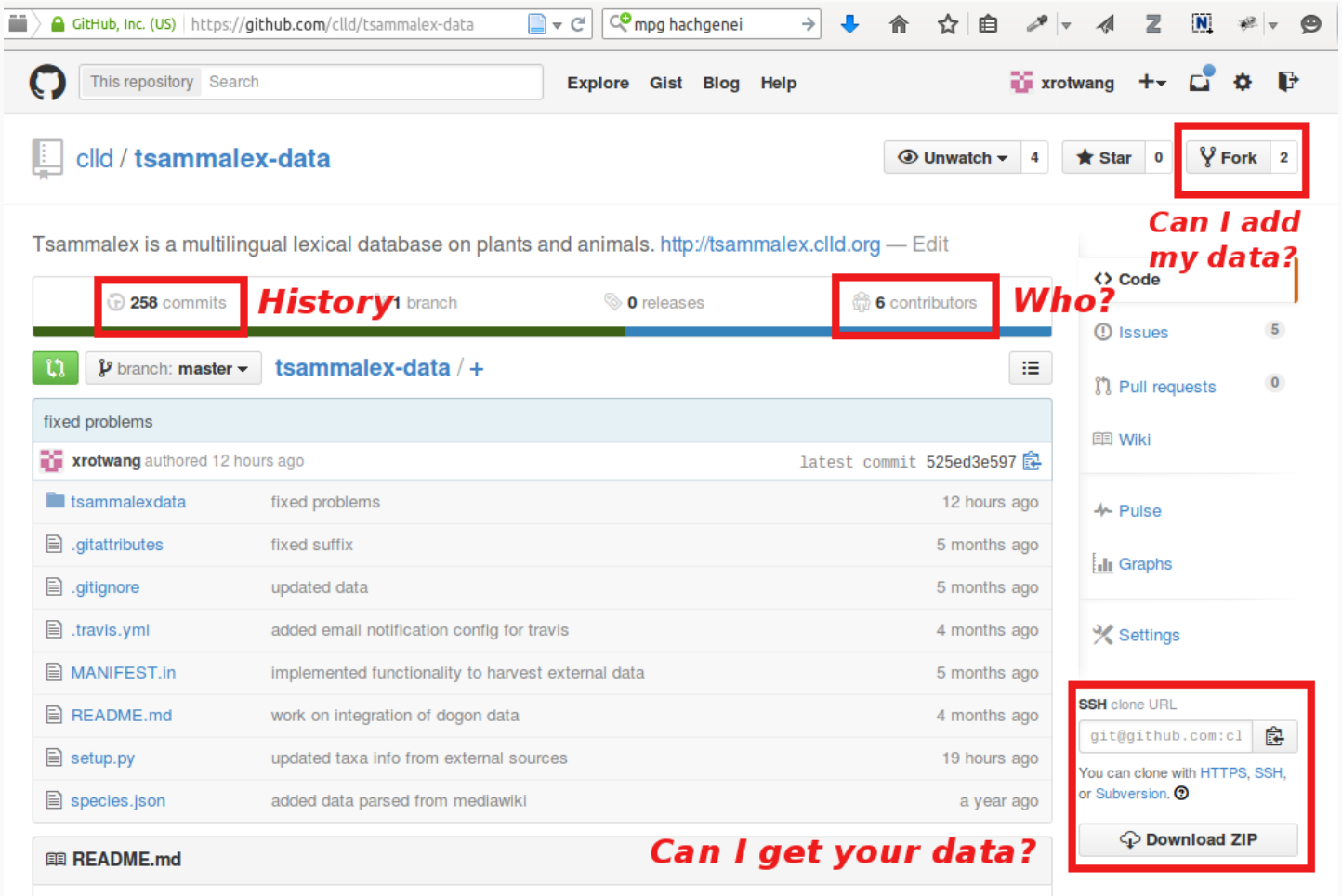


Figure: The data for Tsammalex is curated at clld/tsammalex-data.

A DATABASE JOURNAL

Overview

This can be easily extended to a platform for data journals:

cldf	the submission guidelines
pull request	submission with open review
merge	accepting a submission
release	publication

Example: WALS as database journal

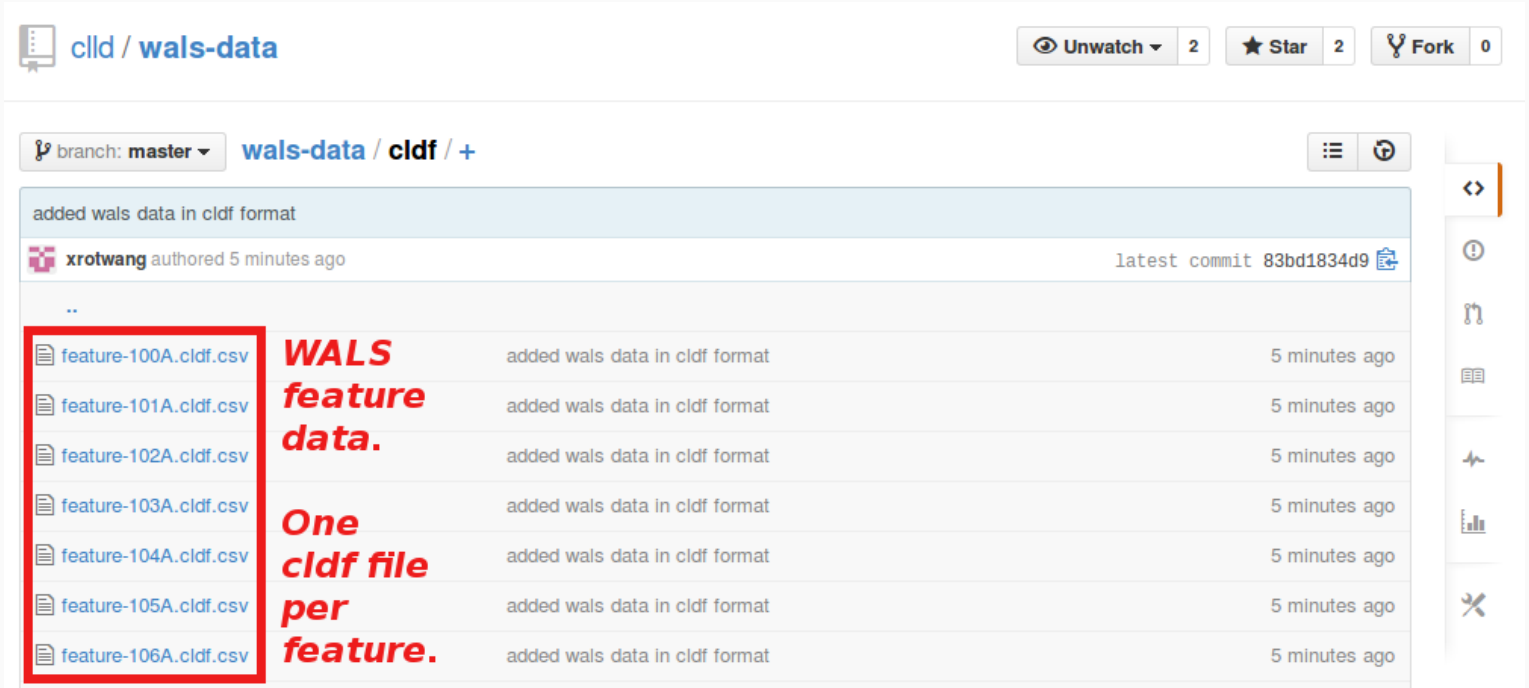
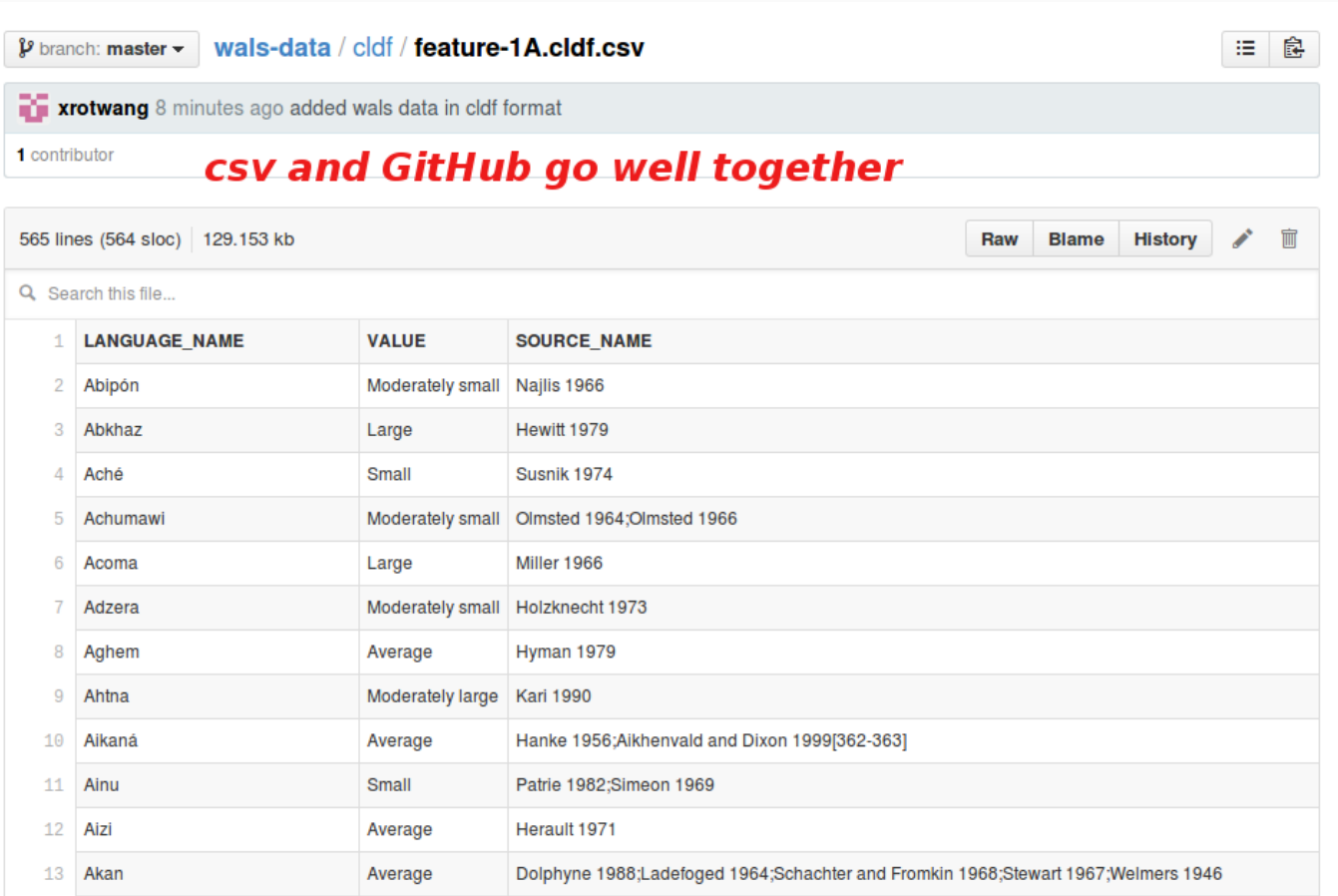


Figure: WALS as database journal.

Browsing cldf on GitHub



	LANGUAGE_NAME	VALUE	SOURCE_NAME
2	Abipón	Moderately small	Najlis 1966
3	Abkhaz	Large	Hewitt 1979
4	Aché	Small	Susnik 1974
5	Achumawi	Moderately small	Olmsted 1964;Olmsted 1966
6	Acoma	Large	Miller 1966
7	Adzera	Moderately small	Holzknecht 1973
8	Aghem	Average	Hyman 1979
9	Ahtna	Moderately large	Kari 1990
10	Aikaná	Average	Hanke 1956;Aikhenvald and Dixon 1999[362-363]
11	Ainu	Small	Patrie 1982;Simeon 1969
12	Aizi	Average	Herault 1971
13	Akan	Average	Dolphyne 1988;Ladefoged 1964;Schachter and Fromkin 1968;Stewart 1967;Welmers 1946

Figure: GitHub supports tabular data well.

PULL REQUESTS AS SUBMISSIONS

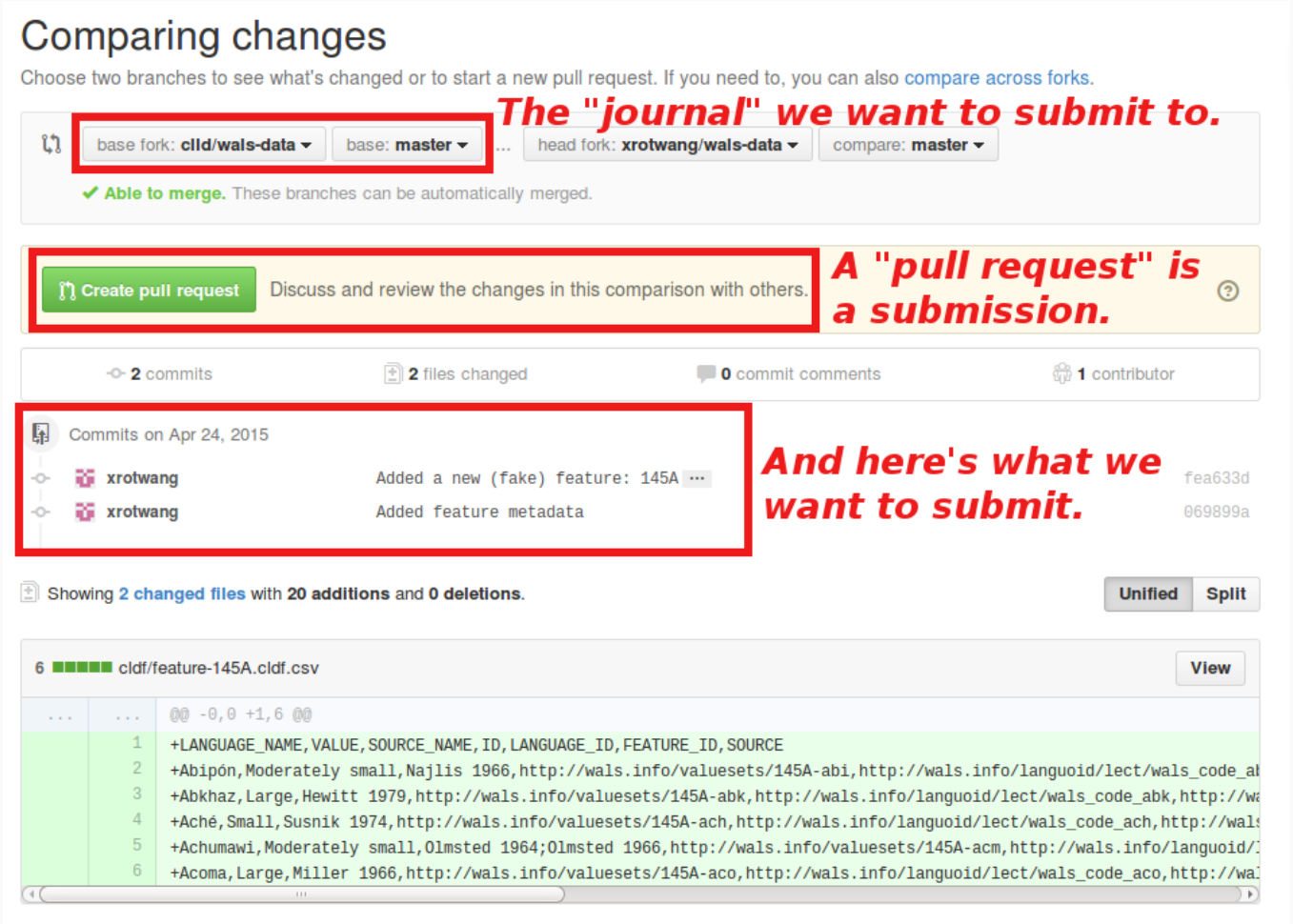


Figure: *pull requests* as submission mechanism.

LIST OF SUBMISSIONS

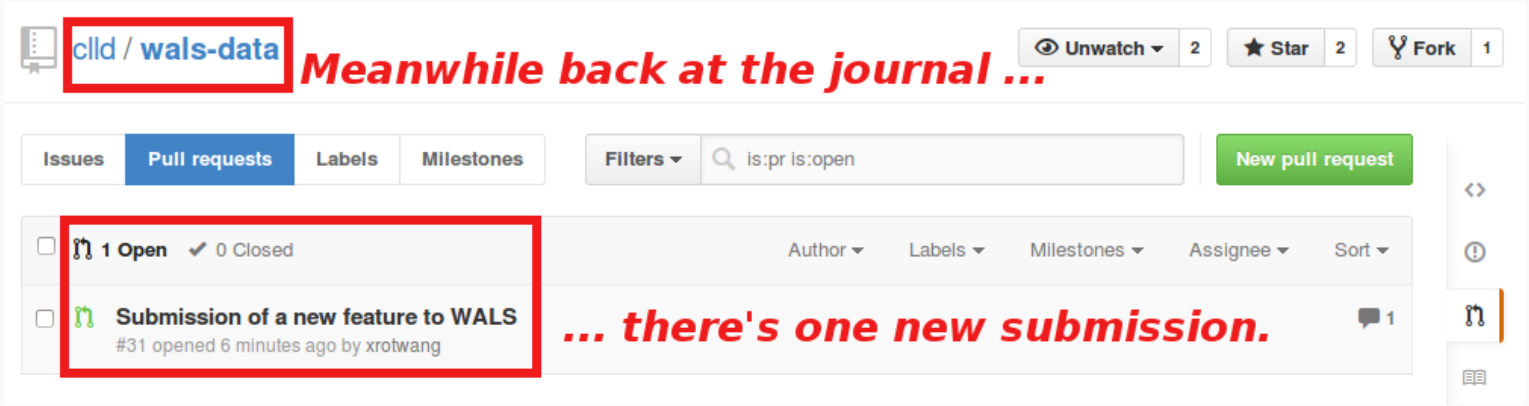


Figure: The list of open pull requests serves as the backlog for the editors.

OPEN REVIEW

Figure: Reviewers can comment on pull requests.

ADDRESSING REVIEW COMMENTS

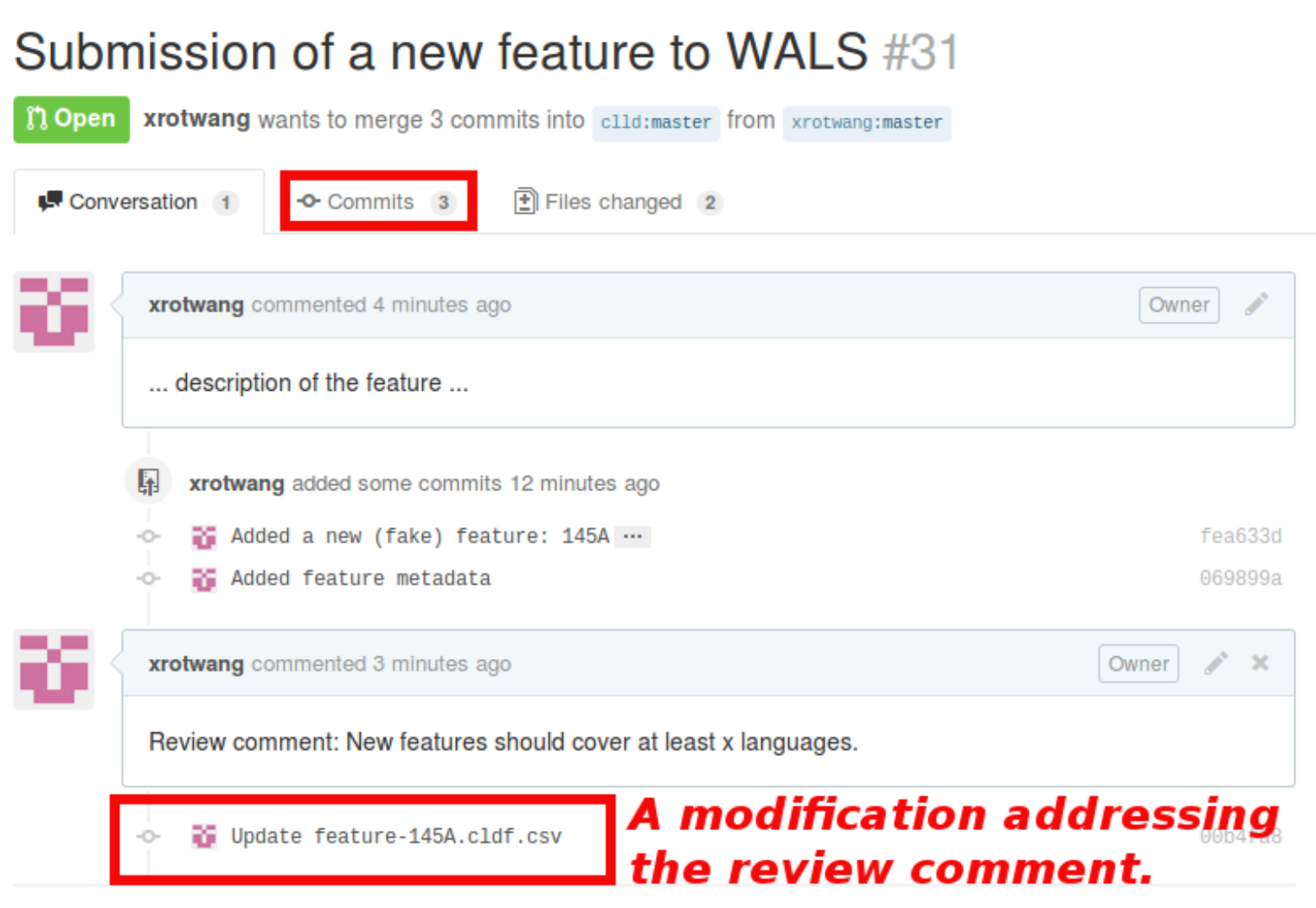


Figure: Review comments can be addressed by adding commits to the pull request, i.e. amending the submission.

Submitting language surveys

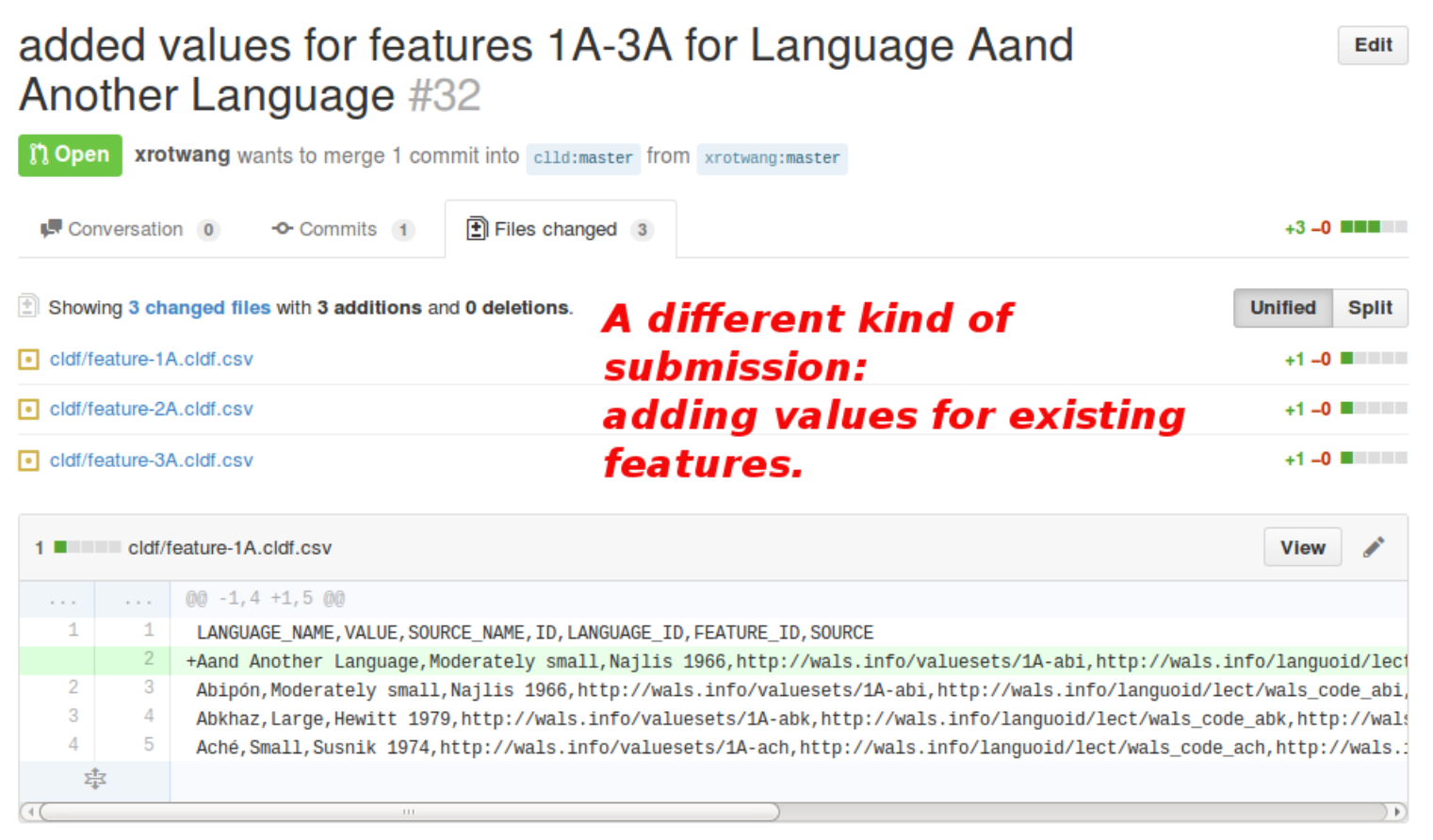


Figure: Since pull requests can bundle changes to many parts of the repository, allowing language surveys in addition to new features as submissions is not a problem.

BELLS AND WHISTLES

Blame

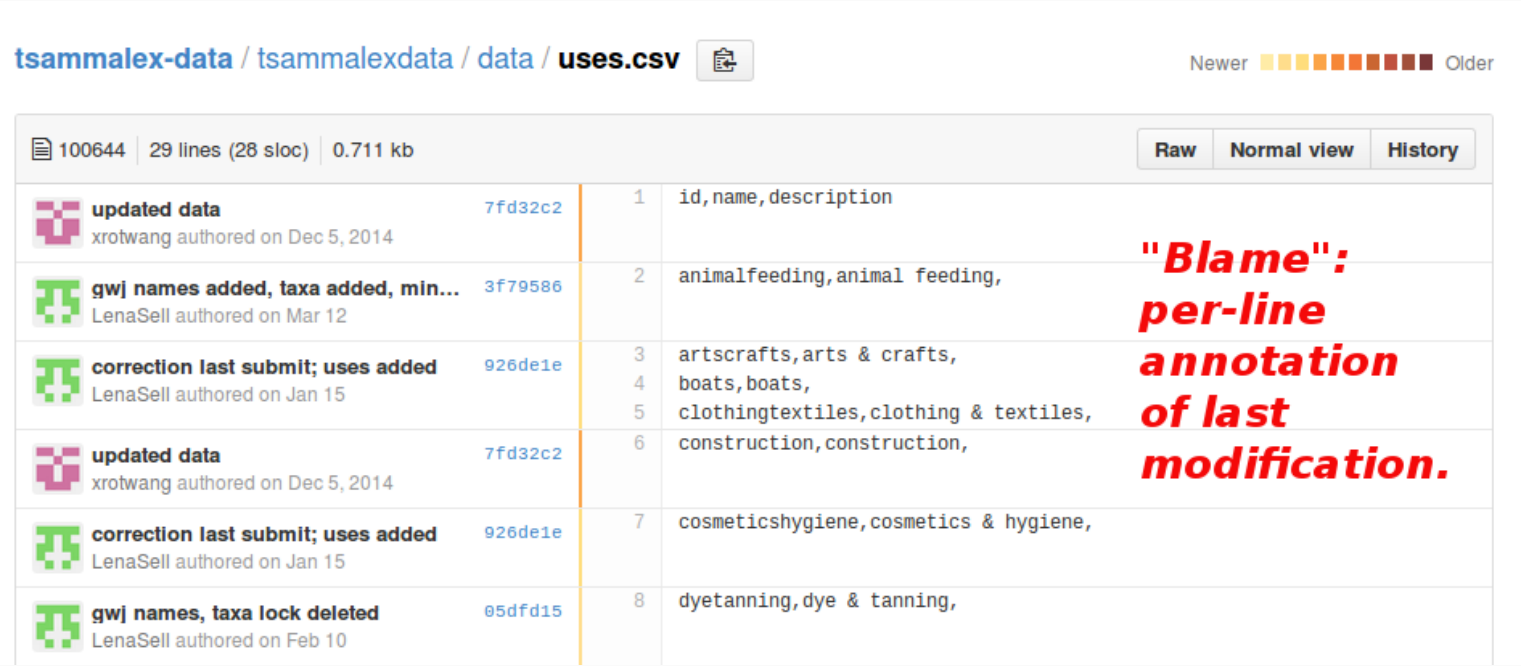


Figure: The Git *blame* functionality provides per-line annotation of last modification - thus allowing provenance tracking not only on file level.

Continuous integration

Let's go further borrowing best practices in software development.

Continuous integration

> *In addition to automated [...] tests, organisations using CI typically use a build server to implement continuous processes of applying quality control in general — small pieces of effort, applied frequently.*
>
> – http://en.wikipedia.org/wiki/Continuous_integration

CI for GitHub

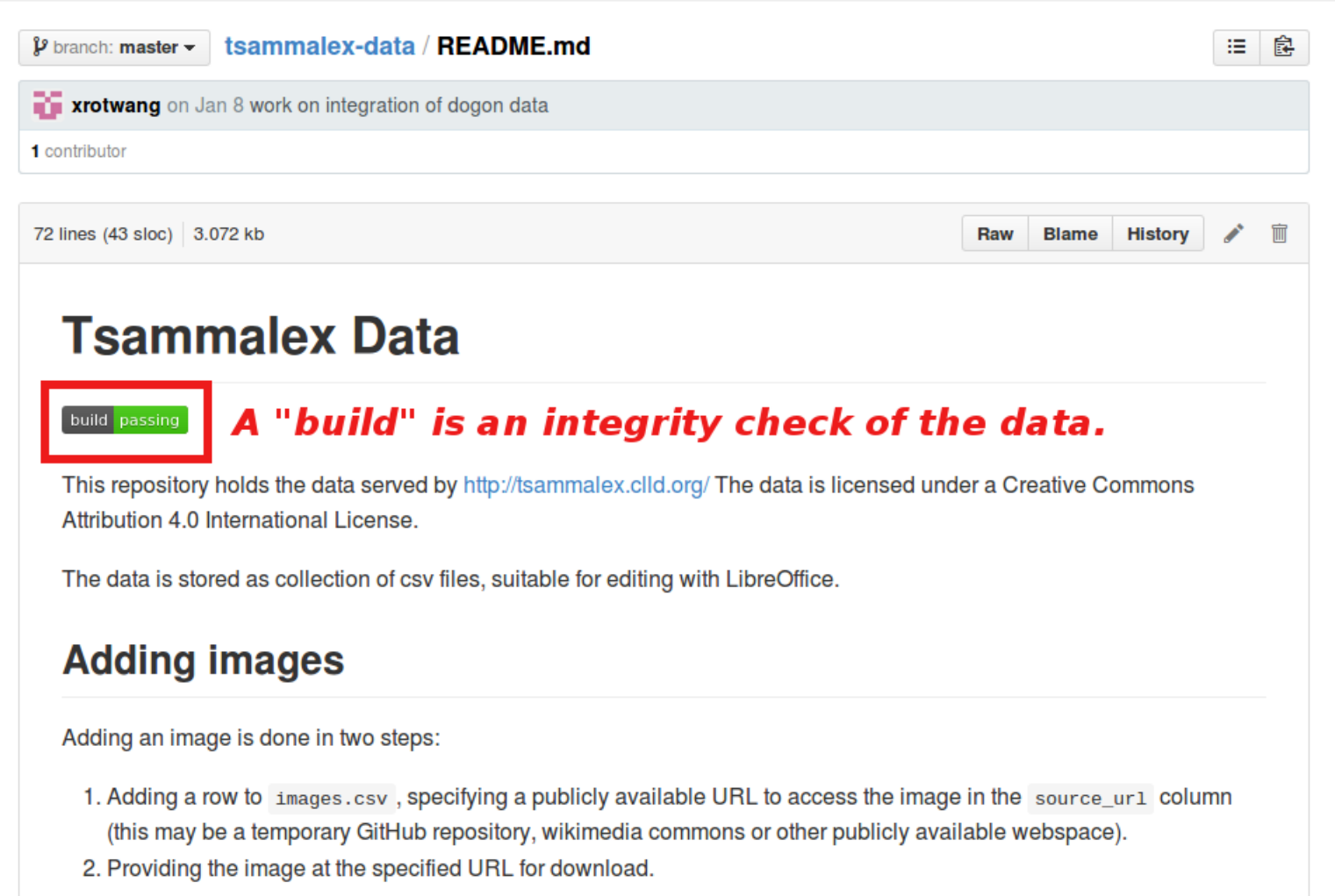


Figure: GitHub repositories can be registered with CI service provider Travis-CI.

CI Build history

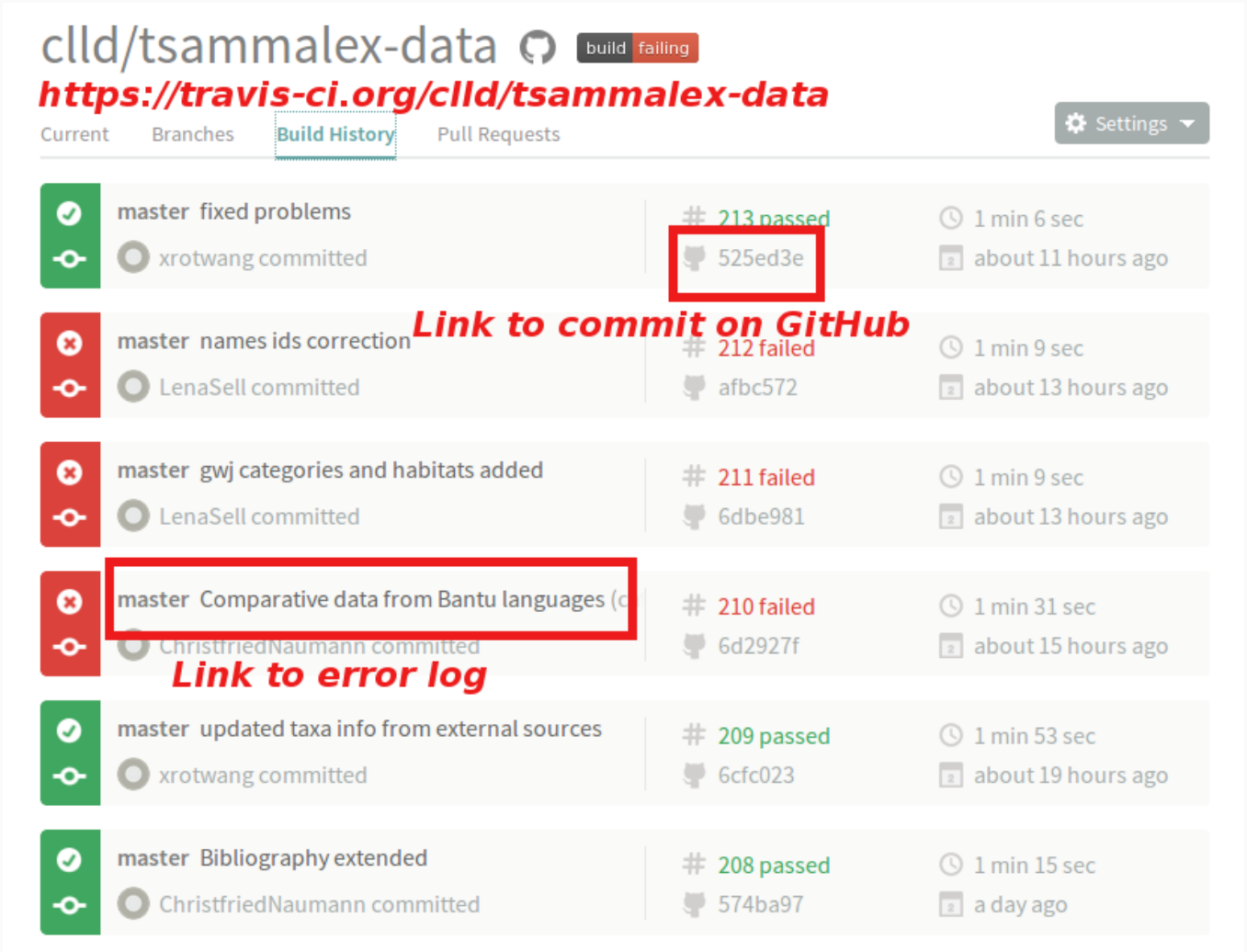


Figure: The build history relates builds and repository changes.

CI BUILD LOG

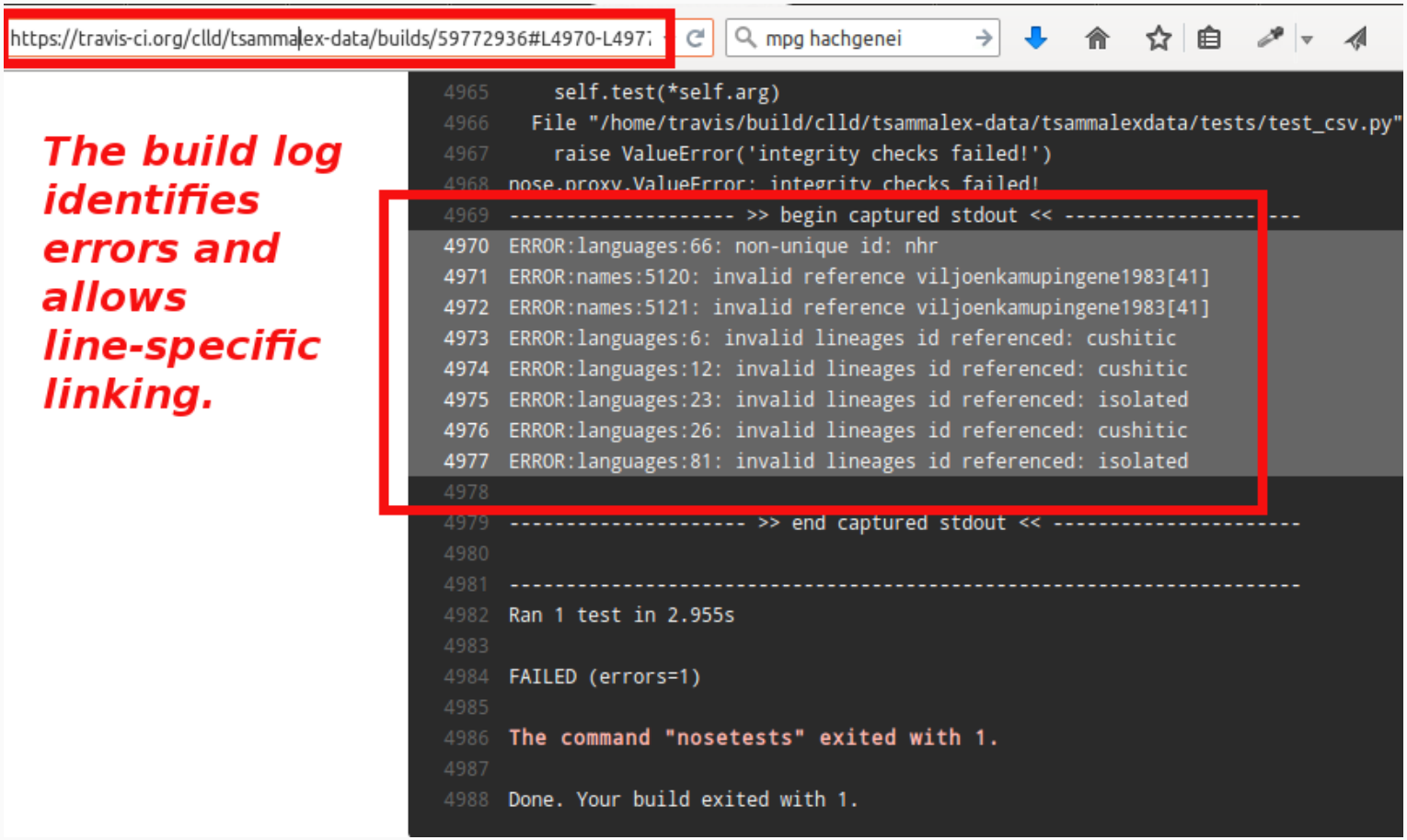


Figure: Build log for error reporting.

CI: ADDRESSING BUILD ERRORS

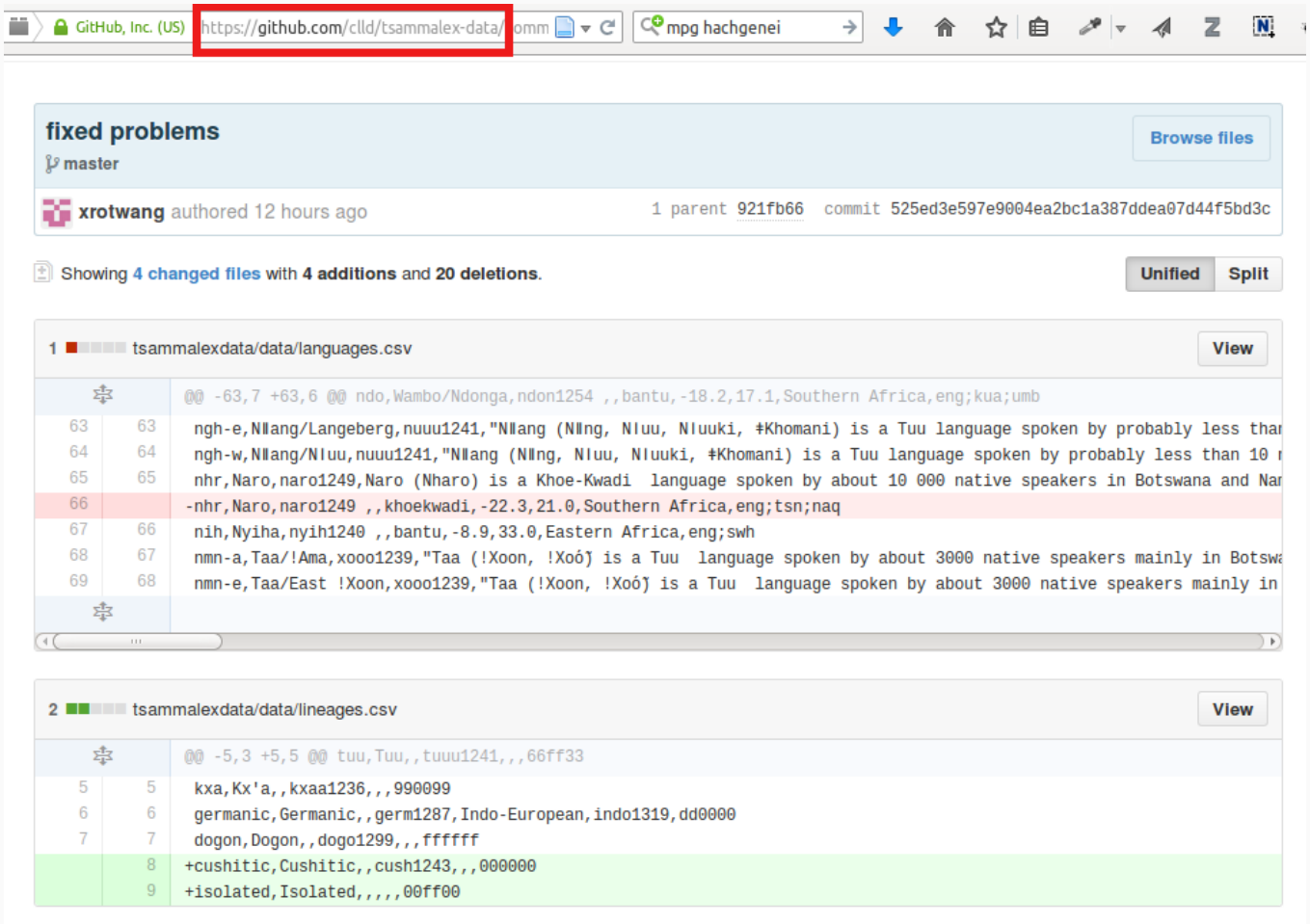


Figure: The URL to the build log could be used in a commit log to link changes back to the error report.

But what if GitHub ...?

Does this introduce too much dependence on GitHub**.com**?

There are some mitigating factors:

- git is a *distributed scm*, thus each clone contains all the data!
- There are alternative git hosting platforms like BitBucket.
- and then there's ZENODO

ZENODO

ZENODO solves the longterm preservation and citability issue for GitHub repositories by

- archiving releases (”issues”) of GitHub repositories
- assigning a DOI to each release

Glottolog 2.4 at ZENODO

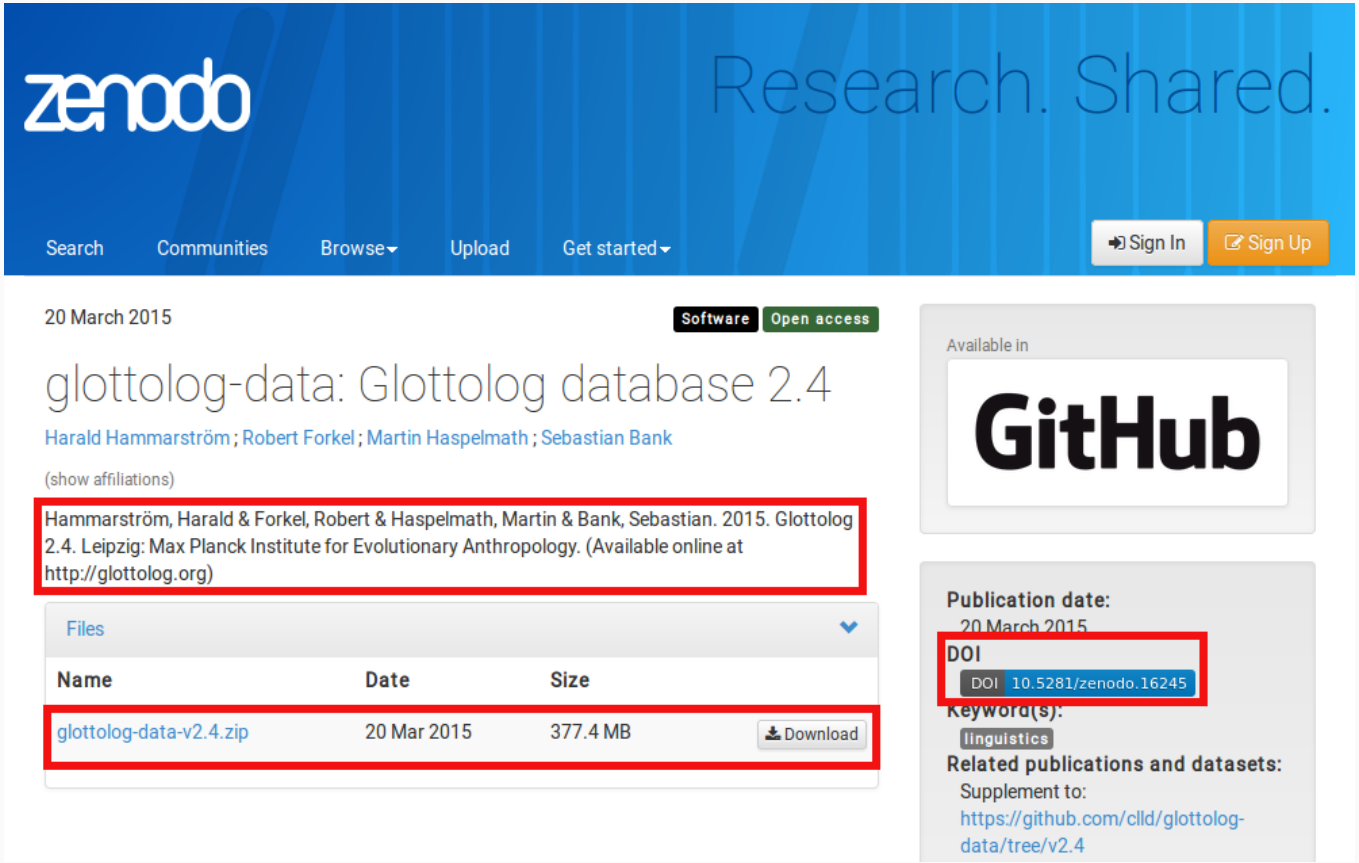


Figure: http://dx.doi.org/10.5281/zenodo.16245

Summary

If your data is code, treat it as such.

And yes, GitHub is the missing editorial backend of `clld :)`.

`clld.org`