

## WebApollo: A Web-Based Sequence Annotation Editor for Community Annotation

Ed Lee<sup>1</sup>, Gregg Helt<sup>1</sup>, Nomi Harris<sup>1</sup>, Mitch Skinner<sup>2</sup>, Christopher Childers<sup>3</sup>, Justin Reese<sup>3</sup>, Jay Sandaram<sup>3</sup>, Christine Elisk<sup>3</sup>, Ian Holmes<sup>2</sup>, Suzanna Lewis<sup>1</sup>

<sup>1</sup>Berkeley Bioinformatics Open-source Projects, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA

<sup>2</sup>Department of Bioengineering, University of California at Berkeley, Berkeley, CA, 94720, USA

<sup>3</sup>Georgetown University, Washington DC, 20057, USA

Contact: Ed Lee (elee@berkeleybop.org)

As technical advances make sequencing faster and cheaper there are more and more community annotation efforts, augmenting the traditional centralized model where curators for a given genome project are all at the same physical location. This trend is particularly strong for smaller genome projects that largely rely on contributions from geographically dispersed community experts. WebApollo was designed to provide an easy to use web-based environment that allows multiple distributed users to edit and share sequence annotations.

WebApollo is comprised of three components: a web-based client, a server-side annotation editing engine, and a server-side service for providing the client with data from different sources, including databases at the University of California at Santa Cruz, Ensembl, and Chado.

The web-based client is designed as an extension to JBrowse, a Javascript-based genome browser that provides a fast, highly interactive interface for visualizing genomic data. This JBrowse extension provides the gestures needed for editing annotations, such as dragging and dropping features to create new annotations, dragging to change boundaries of existing annotations, and using context-specific menus for modifying features. The extension offers access to the annotation-editing service and the data-providing services as well.

The server-side annotation-editing engine is written in Java. It handles all the logic for editing and deals with the complexities of modifications in a biological context, where a single change (for example, splitting or merging transcripts) can have multiple cascading effects. Edits are stored persistently in the server, allowing users to quickly recover their data, should either their browser or the server crash. The server makes use of the Comet model to provide synchronized updating over multiple browser instances, so that if one user edits an annotation, anyone who is viewing that annotation sees the changes instantly in their browser window.

The server-side service for providing data to the client is built on top of Trellis, a DAS server framework. It uses format injection to provide JBrowse-supported JSON data structures, rather than the more verbose DAS XML. We also developed a Trellis plugin to access data from the UCSC MySQL genome database, which provides quick access to that popular data source.

All three components are open source and provided under the BSD License.

Source code and demo:

<https://github.com/berkeleybop/jbrowse> (client side code)

<http://code.google.com/p/apollo-web> (annotation editing engine server code)

<http://code.google.com/p/gbol> (data model and I/O layer code used by edit engine)

<http://code.google.com/p/genomancer> (Trellis server code)

<http://icebox.lbl.gov:8080/ApolloWebDemo> (demo)