WebApollo
extending JBrowse to support DAS
&
genomic annotation editing

Gregg Helt, Ed Lee, Nomi Harris, Mitch Skinner, Suzanna Lewis, Ian Holmes
Lawrence Berkeley National Laboratory & UC Berkeley
Main Goal of WebApollo

Provide an interactive visual editor for genome annotation and curation, that anyone can use via a standard web browser
Build on existing infrastructure

- For front end start with current version of JBrowse (AJAX-based genome browser)
- Port visual editing features of Apollo (Java-based genome curation tool) to JBrowse
- For access to non-editable annotations, do whatever it takes to enable WebApollo to access as many types of genomic data as efficiently as possible (DAS?)
- Data access implementations and ideas from Integrated Genome Browser (IGB) and Trellis DAS framework
- For back end to editable annotations and curations, start with current Apollo backends (mostly GMOD Chado databases)
- Evolve later to allow more back end flexibility (DAS writeback?)
JBrowse

• Mitch Skinner & Ian Holmes, UC Berkeley
• JavaScript Genome Browser
• Version 1.2 just released
• CSS & DIV based rendering for interval data
• Tiled Images for quantitative data
• Nested Containment Lists for speedy interval queries
• jbrowse.org
• WebApollo uses and extends Jbrowse code base
Apollo

• Suzi Lewis, Nomi Harris, Ed Lee, etc., LBNL
• Desktop Genome Annotation Editor
• Implemented in Java
• GMOD Chado database back end
• WebApollo using revised back end GBOL (Java model of Chado schema)
Whatever it takes to enable WebApollo to access as many types of genomic data as possible

• More practically, efficient access to public data from UCSC, Ensembl, and GMOD Chado databases
• Unified strategy like DAS is preferred
• Leverage efficient JBrowse JSON data formats and nested containment lists
• Solution: Trellis, a modular DAS2 server that supports alternative content formats
• Trellis plugins to proxy for other DAS servers
• Poka Trellis plugin for direct UCSC database access
Alternative Content Format Support (Format Injection)

Data Source → Capabilities Plugin → Model to Data Source Query → Data Source to Model Response → Model to Data Model → DAS Data Model

Trellis Framework Servlet

- URL/HTTP Request to Model
- Model to DASXML HTTP Response
- Model to NCLList JSON HTTP Response
- Model to BED HTTP Response
- Model to ProtoBuf HTTP Response

Data Source to Model Response

JBrowse Client
DAS $\rightarrow$ JBrowse Proxy
Ivy Plugin + Trellis + Format Injection
UCSC → JBrowse Proxy
Poka plugin + Trellis + format injection
WebApollo Demo
WebApollo Demo2
• All open source
• Client Code on GitHub: berkeleybop/jbrowse
• Server Code all over the place
  – Trellis: http://code.google.com/p/genomancer/
  – GBOL: http://code.google.com/p/gbol/
  – WebApollo-specific server bits:
    • http://code.google.com/p/apollo-web/
  – Etc.