Gramene 2.0 Ideas

Ken Youens-Clark

Lab Talk July 10, 2012

Code Repository

- Moving from Subversion to Git
- Consistent with other lab projects
- More flexible, modern, cool
- Will use more divisions, e.g., for web site, pipelines/build code, etc.
- Where should it be hosted? Github?

Server-side Language(s)

- Probably best to stick to Perl as we're so dependent on Ensembl API
- Thoughts about Nodejs, but is it just flash in the pan? Where could server-side Javascript be useful?

Web Framework, REST

- Catalyst vs Mojolicious; very interested in PSGI/Plack as it seems to provide a much-needed way to test/stress website programmatically, outside of a web server
- Other contenders?
- Absolute necessity to create and use REST for all data requests; want a framework that makes that eas(y|ier)

Web Server

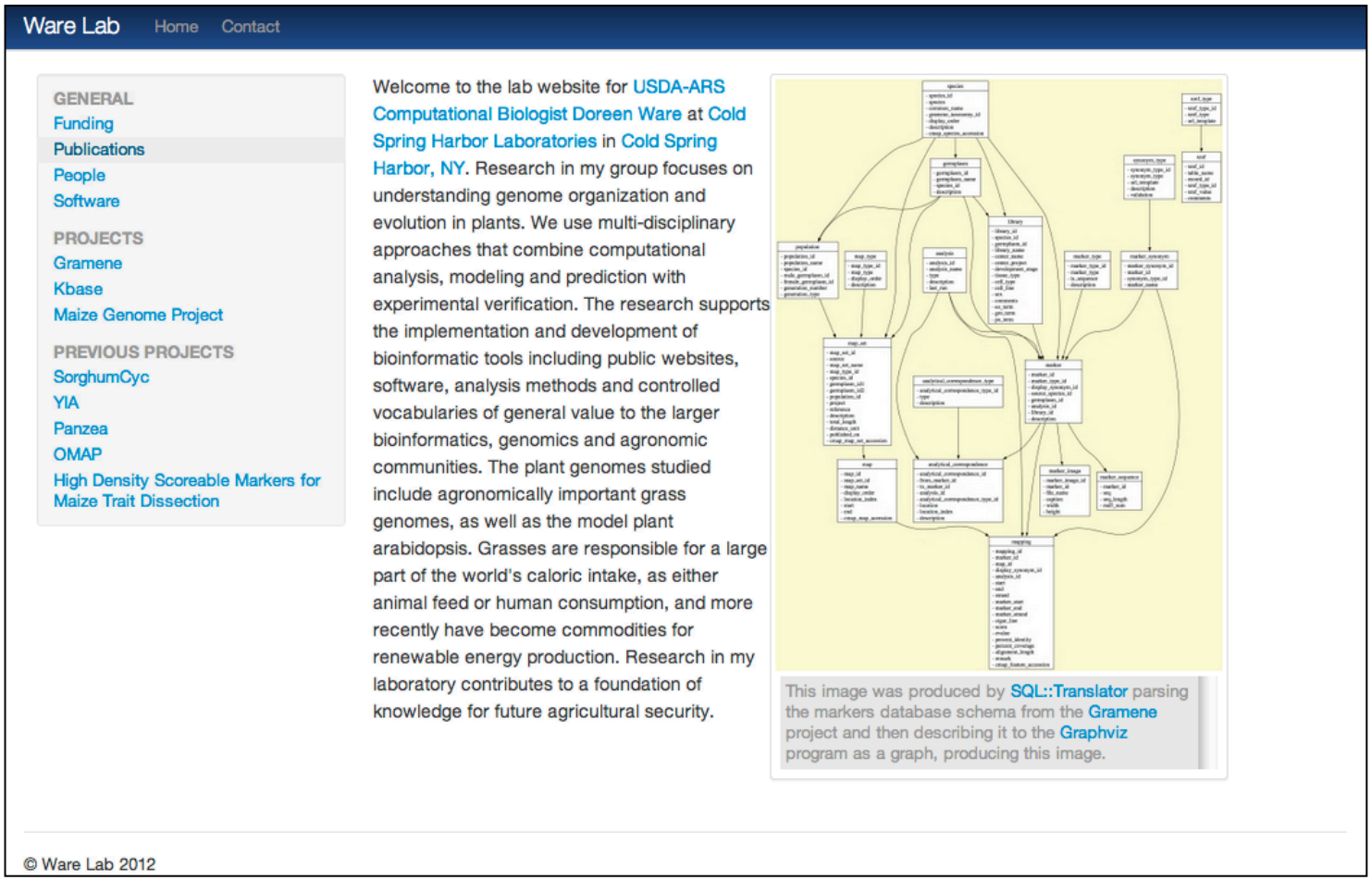
- Stick with Apache/mod_perl?
- Nginx

Misc

- Module::Build vs Module::Install; really interested in the “inc::Module::Install” to include M::I in dist, don’t need to install anything first
- Use something like “local::lib” to install all needed modules into a “gramene/lib”? Include with dist or install from CPAN (and worry about version conflicts)?
- Require a recent version of Perl (5.12)

Look and Feel

- Move to Bootstrap? For what parts?
- Should Gramene just be a skin for Ensembl like maizesequence.org?



Wiki for Static Content?

- About 2500 static files in “html” dir
- How much can we lose?
- The Gramene::Page to inject headers is crude, but works
- Wiki would gain automatic search

Quick Search

- Currently ~2.8M records using MySQL FULLTEXT indexing
- Somewhat slow
- Would like to explore flat-files indexed with Apache Lucy?
- Goal is to get users to Ensembl views ASAP (a la Google) or to some sort of Gramene view of objects?

Markers, Mappings

- Markers and CMap queries are most common MySQL killers
- Move all mappings into flat files:
  - GFF for chr:start-stop queries, probably served by DAS (Proserver using tabix) even for REST, ready for FTP
  - Custom FastBit file for queries based on marker_id, marker_type_id, etc.
- Maybe stick with MySQL for identifiers (names, synonyms)

Proserver, DAS

- Have written GFF_Tabix.pm SourceAdaptor module to easily handle GFF files indexed with tabix
- While working on maize transposon data for Lifang, Andrea; display didn't work in our Ensembl until this release
- Maize expression tracks, data from Bremen (MaizeGDB)?
- Can base new Proserver code directly on the GFF dumps from alignment pipeline, obviate separate build for this, simplify our configuration (not exactly sure how we'd handle current tracks like "Brachy ESTs aligned to rice MSU7")

CMap

- Will something come from Iris that could show side-by-side map views?
- Andrew's views of gene build comparisons?
- Put markers, CMap dbs on separate db server (flume, gaining a new server sometime soon)

SSWAP

- http://sswap.info/
- Attended SSWAP workshop in Santa Fe last month
- Can work locally with Damian Gessler to integrate SSWAP into Gramene
- REST API must come first

Variation

- I'm to be doing some work with Ensembl variation?
- I've stopped indexing variation in quick search as just one species can add 16M records
- What/how, if anything, do people want to search on (IDs, traits, just locations)
- Integrate SNP Query into gene views, Xrefs, ontology views?

Xrefs

- Xref pipeline was pretty tragic last run
- Needs fairly major update based on API changes
- More importantly, we need to ensure all Gramene-specific xrefs are going in properly and make it into Mart
- Deprecate any Gramene view that can be supported via xrefs?

Ontologies

- Possibly segregate ontologies into separate dbs instead of integrating into one?
- Allows updating of individual ontologies, e.g., PO that updates most often
- Keep GAZ? We don't use it anywhere.

Deprecation

- What can we lose (static content, genes and proteins modules)

Testing

- Need a test suite to run, esp. to test release
- Have created a couple scripts for testing, e.g., all db connections to ensure all in place and can connect with credentials (work here to seamlessly integrate all Ensembl dbs from “ensembl.registry”); another checks that the configuration file is sane