Project Acronym: MuG

Project title: Multi-Scale Complex Genomics (MuG)

Call: H2020-EINFRA-2015-1

Topic: EINFRA-9-2015

Project Number: 676556

Project Coordinator: Institute for Research in Biomedicine (IRB Barcelona)

Project start date: 1/11/2015

Duration: 36 months

Deliverable 3.2: A browser-track that implements and connects all the 1D data from a genome or a genomic domain

Lead beneficiary: Institute for Research in Biomedicine (IRB Barcelona)

Dissemination level: PUBLIC

Due date: 31/10/2016

Actual submission date: 15/11/2016

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 676556.

Document history

Version	Contributor(s)	Partner	Date	Comments
0.1	David Castillo	CNAG-CRG	30/10/2016	First Draft
0.2	Mike Goodstadt	CNAG-CRG	3/11/2016	Second Draft
0.3	Marc A. Martí-Renom	CNAG-CRG	4/11/2016	Third Draft
0.4	Mike Goodstadt	CNAG-CRG	5/11/2016	Final Draft
0.5	Mark McDowall	EMBL-EBI	10/11/2016	Minor spelling corrections
0.6	Marco Pasi	UNOT	11/11/2016	Minor spelling corrections
0.7	Mike Goodstadt	CNAG-CRG	15/11/2016	Updated TADkit browser URL
1.0				Final Version Approved by Technical and Supervisory Board

Table of contents

Executive summary

One of the basic components of the MuG visualization tool (the TADkit browser http://sgt.cnag.cat/3dg/mug/tadkit) is the representation of 1D genomic datasets. That is, the information encoded in the genome. This document introduces the development of the part of the MuG browser rendering such lineal information of the genome. It also addresses the convenience of using existing tools for the visualization of lineal tracks while focusing on its connection with new developed 2D and 3D data components.

1 INTRODUCTION

Genomic data has been historically characterized and interpreted as linear sequences of base-pairs. This 1D data logically forms the core horizontal visualization in most tools developed to explore the genome. Additional data are then aligned to genomic coordinates of such browsers and stacked to form easily comparable tracks.

The TADkit browser app is being designed to visualize data that cannot be represented as conventional tracks (see D3.1). However linear tracks remain key to genomic analysis both for familiar orientation and to inform exploration of the new data. Tracks therefore remain an essential component of the TADkit browser to provide effective data access, a coherent user experience, and integration with novel visualizations.

2 REQUIREMENTS

2.1 Effective Data Access

There are a number of standard file formats which are generated by tools and institutions across the world. These store data from global sequencing initiatives, research from individual labs and analytical processed data from various institutes. The resulting files typically flat human-readable text files of immense size (GB range) normally provided compressed. Genome browsers have to efficiently fetch, display, filter and navigate ranges of data, avoiding loading delays for the user.

To maximize the speed and coverage there are different strategies in the manner how data is stored and indexed. These are technically difficult but there are existing solutions that have already implemented this process of optimization and which can be divided in four major categories depending on how the browser accesses the data:

2.1.1 Web Services

Web Services gather, process and store data to make them available through online APIs. The browser accessing the web service must be configured to communicate with the services *endpoints* and retrieve data in the proper format. Previously, this was coordinated via the now defunct Distributed Sequence Annotation System (DAS) web service protocol. Current endpoints are Ensembl, OpenCGA, CellBase, as well as other smaller data providers. The key advantage of these services is data streaming to reduce network load and end-user wait times.

2.1.2 Track Hubs

Track Hubs are a metadata standard created by UCSC which enables the easy referencing of data located anywhere across the Internet. The data remains where is originally provided, and small text files contain the metadata indicating the data source. These can then be added to official public registries for ease of discovery and reference. In some cases, the information is fragmented and indexed to optimize the streaming and rendering of tracks.

2.1.3 Remote data via HTTP/FTP files

Without using Track Hubs, data can be hosted on publicly accessible servers for direct download in their entirety. This is often used by researchers when supplying supplementary data for publications. However, such large downloads may be inefficient, increasing network load, waiting times for users. They also make research vulnerable to data loss due to maintenance issues, reducing discoverability and reproducibility.

2.1.4 Local files or database

Access to locally stored data in files or databases is an essential part of the process of research and therefore must be a core feature of any browser. Simple file selection can now be enhanced with facility to drag-and-drop files into web applications.

2.2 A Coherent User Experience

There has emerged a consistent visual grammar and range of track types for genomic data as seen in the tracks of common genome browsers. However, they vary greatly in methods of implementation, graphical styles and user interface. All solutions examined here render to Canvas or SVG but there is as yet no common standard nor library of visualization elements. The ideal tool should take advantage of high-resolution screens, high data-pixel ratio and a modern use-case driven interface design.

2.3 Integration with Novel Visualizations

Many genome browsers use older programming languages not easily adapted to new user devices, evolving web-based use or advances in bioinformatics. Tool development must avoid obsolescence by choosing well-supported, road-mapped, widely adopted technologies. Therefore, the solutions examined here are web-based, client-side and coded in JavaScript. Ideally, they should be convinced as web-components, have minimal dependencies and permit creation of plug-ins.

3 WEB-BASED TRACK BROWSERS

3.1 Existing Tools

The state-of-the-art web technologies used in app development facilitates the creation of TADkit specific tracks. However, there are a number of existing track browsers which are open-source, portable and extensible and are used by the scientific community [1].

3.1.1 JBrowse

JBrowse is the most widely used web-based browser, maintained since 2009 by the GMOD community [2][3]. It is well-documented, easy to customize and can access various data sources. However, it is difficult to integrate as data must be chunked and indexed as a hierarchy of JSON files rather than using web services or databases directly. The dominant header menu and cluttered navigation bar impede seamless integration yet the browser is fully featured with a complete range of track types.

3.1.2 BioDalliance

BioDalliance, funded by the Wellcome Trust (2011) and BBSRC (2013-2014), is an innovative state-of-the-art browser [4][5]. Originally using the then predominant DAS protocol, it can now access Track Hubs, Ensembl and many file types. It is written in clear, native JavaScript and so is easy to extend and plug-in to. The neutral and elegant design facilitates unobtrusive embedding in websites and apps.

3.1.3 *Genome Viewer (JSorolla API)*

The JSorolla genomic tools were built for the CellBase RESTful data service app, now maintained at Cambridge University [6][7]. Apart from the CellBase site (that is, Genomemaps.org) this API is core to a number of other CellBase-based services. It uses Polymer web-components so that the Genome Viewer component can be used independently, sourcing data from CellBase, OpenCGA and Ensembl services. Use of non-native JavaScript dependencies, lack of Canvas rendering and strong visual identity may complicate embedding.

3.1.4 *Genoverse*

Genoverse was developed for DECIPHER and Ensembl and funded by the WTSI [8][9]. It can fetch data not only from Ensembl but also from a wide range of online or local sources. It does not use modern web technologies, generates track images server-side and has an idiosyncratic interface, however the code is native JavaScript and so easy to adapt. The EBI are currently in the process of full adoption of Genoverse for their web-service.

3.1.5 *IGV.js*

IGV.js is the most recent addition to this collection although it is based on the popular stand-alone genome browser IGV [10][11]. Although nascent it is well-supported, well-coded modern JavaScript. Being an early release it is lacking more complex navigation features but has a simple, clean frontend. It is used by the Global Alliance for Genomes and Health and by Google in their online Genomics suite. As it was released in September 2016 we are still reviewing it but recommend keeping a close watch on its development.

3.2 Tools Summary

The following table summarizes the existing tools:

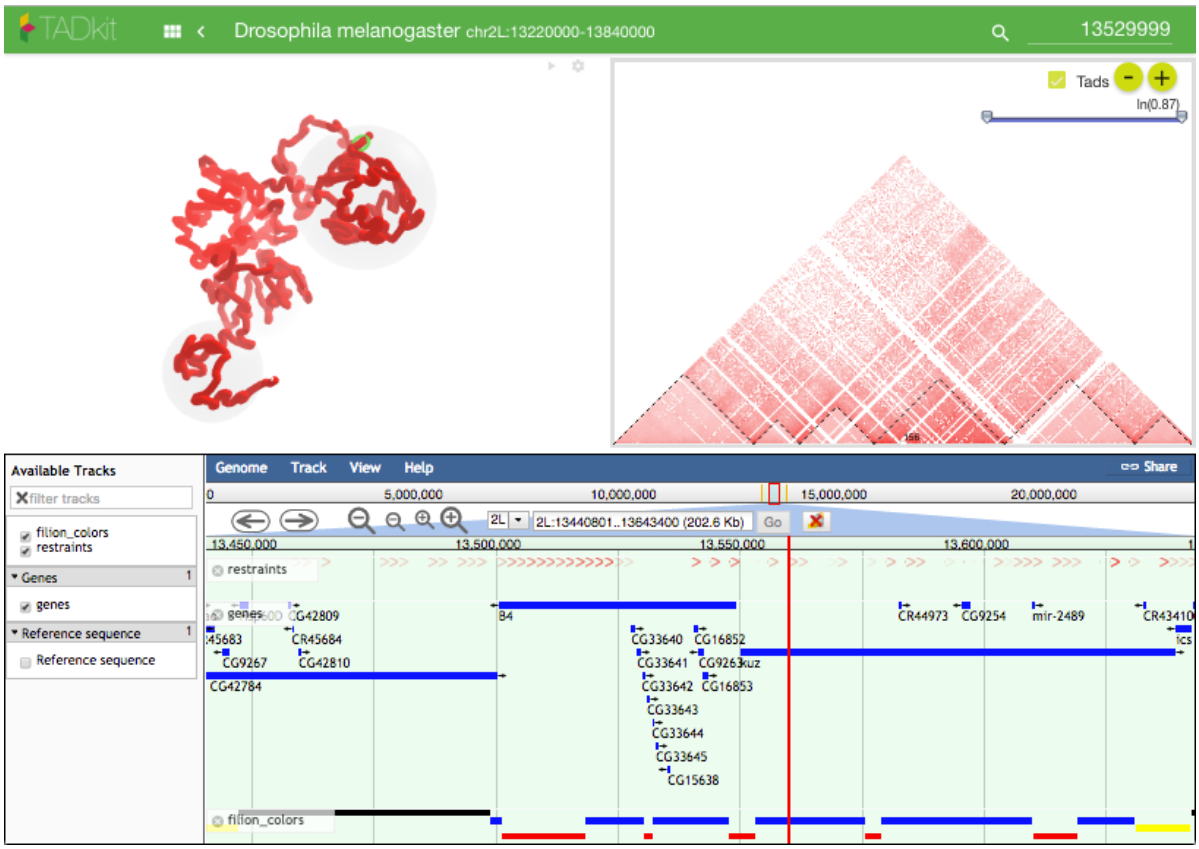
	Web Services access	Process Track Hubs	Remote Data access (HTTP/FTP)	Local Data from File or Database	Coherent / Modern / Use-case	Maintained / Extensible / Components	Adoption
JBrowse	None	No	✔ [1]	✔ [2]	_ _ ✔	✔ _ _	NCBI Widespread in research & webs
BioDalliance	Ensembl JBrowse	✔	✔	✔	✔ ✔ ✔	_ ✔ ✔	GENCODE UK10K
Genome Viewer (Jsorolla API)	Ensembl, OpenCGA, CellBase	✔	✔	✔	✔ ✔ ✔	_ ✔ ✔	Genomemaps EBI 100,000 Genomes
Genoverse	Ensembl	In roadmap	✔	✔	✔ _ _	✔ ✔ _	Sanger Institute EBI
IGV.js	GCS [3] CA4GH [4] Ensembl [5]	✔	✔	✔	✔ ✔ _	✔ ✔ ✔	Google Genomics Global Alliance

[1] Data can be displayed if downloaded in full but will lack JBrowse conversion for optimization.
[2] Conversion to a JBrowse chunked JSON dataset is expected although local file can be displayed directly.
[3] Google Cloud Storage
[4] Global Alliance for Genomics and Health
[5] Requires plugin development

4 TRACK TOOL TESTING

4.1 A JBrowse plugin

Due to its popularity among the scientific community and the MuG partners, it has been chosen to test the integration of JBrowse within TADkit. JBrowse can only be embedded as an iframe in third party applications. An iframe is not the most convenient solution for integrating applications because it complicates the communication between the different modules. JBrowse can be further extended by the development of plugins, so given the iframe issues, TADkit has been integrated as a plugin. The advantage of developing the integration in a plugin is that JBrowse core code remains untouched and independent from TADkit. See http://sgt.cnag.cat/3dg/mug/tadkit



5 CONNECTION 1D AND 2D COMPONENTS

Currently we have already implemented a 2D track within TADkit, which allows for an efficient integration of JBrowse as a 1D component together with the rendering of Chromosome Conformation Capture (3C)-like experiments (2D matrix). The integration implies the matching of the genomic regions in both components 1D and 2D as well as the cross talk of any user action in any of the two components. In such framework, the 2D matrix displayed data region is highlighted in the 1D browser and the navigation in the lineal genomic tracks is also identified as a moving point in the 2D component. A further integration will consist in the identification of two interacting loci by clicking on any point of the 2D matrix. The clicking will result in the two marks placed in the genomic positions of the interacting loci being highlighted in the 1D track (and *vice versa*).

6 CONCLUSIONS

The use of existing tools as plugins for the development of TADkit is the right strategy since TADkit benefits from existing technology to which the end-users are already familiar. Moreover, those tools are already optimized for efficiency and portability. Initially, we have chosen to integrate JBrowse with TADkit due to its popularity among the scientific community and the MuG partners. However, further exploration will continue to assess possible advantages of the other browsers over JBrowse for inputting datasets through the RESTful services of external databases. Once implemented, discussions will be engaged with the MuG community to gather different views and perspectives.

7 REFERENCES

1. Wang J, Kong L, Gao G, et al. A brief introduction to web-based genome browsers. Brief Bioinformatics 2013;14:131-43.

2. Skinner ME, Uzilov AV, Stein LD, Mungall CJ, Holmes IH. 2009. JBrowse: a next-generation genome browser. Genome Res 19:1630–1638. 10.1101/gr.094607.109.

3. JBrowse main website: http://jbrowse.org

4. Down TA, Piipari M, Hubbard TJ. Dalliance: interactive genome viewing on the web. Bioinformatics (2011) 27:889-890. PUBMED: 21252075; ukPMC: 3051325

5. BioDalliance main website: http://biodalliance.org

6. Medina I., Salavert F., Sanchez R., de Maria A., Alonso R., Escobar P., Bleda M., Dopazo J. Genome Maps, a new generation genome browser. Nucleic Acids Res. 2013 Jul;41(Web Server issue):W41-6. doi: 10.1093/nar/gkt530

7. Genomemaps main website: http://genomemaps.org

8. Bragin E, Chatzimichali EA, Wright CF, Hurles ME, Firth H V, Bevan AP, et al. DECIPHER: database for the interpretation of phenotype-linked plausibly pathogenic sequence and copy-number variation. Nucleic Acids Res. 2014;42: D993–D1000. doi: 10.1093/nar/gkt937

9. Genoverse demo website: http://wtsi-web.github.io/Genoverse/

10. Helga Thorvaldsdóttir, James T. Robinson, Jill P. Mesirov. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Briefings in Bioinformatics 14, 178-192 (2013).

11. IGV.js documentation: http://igv.org/doc/doc.html