Which genome browser to use for my data ?

July 2019

The development of genome sequencing projects since the early 2000s has been accompanied by efforts from the scientific community to develop interactive graphical visualisation tools, called **genome browsers** or **genome viewers**. This effort has been further intensified with the advent of high throughput sequencing and the need to visualise data as diverse as Whole Genome Sequencing, exomes, RNA-seq, ChIP-seq, variants, interactions, in connection with publicly available annotation information. Over the years, genome viewers have become increasingly sophisticated tools with advanced features for data exploration and interpretation.

We have reviewed seven genome browsers, selected (arbitrarily) for their notoriety and their complementarity: Artemis, GIVE, IGB, IGV, Jbrowse, Tablet, UCSC Genome Browser. The purpose of this study is to provide simple guidelines to help you to choose the right genome browser software for your next project.

Our selection is of course far from being exhaustive. There are many other tools of interest that could have been included: Genomeview, Ensembl genome browser, Savant for example. We noted also the existence of many specialized viewers: WashU for epigenetics, Ribbon for third generation sequencing, (3rd gen), Single cell genome viewer, Asciigenome,... If you are interested in this work and would like to contribute it, feel free to contact us: bilille-taskforce@univ-lille.fr

We invite you to take a tour.

Authors: Franck BONARDI, Loïc COUDERC, Isabelle GUIGON, Jean-Pascal MENEBOO, Pierre PERICARD, Hélène TOUZET (bilille)

1. Identity forms

We have tested seven genome browsers, for which we distinguish two main types of usage:

Local installation: the tool is installed on a local computer or a server in the lab. Normally, those tools can run offline once they have been installed and that all required data has been downloaded.

Public web instance: the tool is freely accessible on the world-wide-web, without any prior installation.

Artemis, IGB, Jbrowse and Tablet are in the first category, while GIVE, IGV and UCSC offer both possibilities. For local install, we also make the difference between native applications and web applications (see below). This means that some tools are available in three versions. Each time it made sense, we analyzed these versions separately.

	Artemis	GIVE	IGB	IGV	Jbrowse	Tablet	UCSC
LOCAL INSTALLATION							
Native app	●		●	●	●	●	
Web app		●●		●●	●●		●
Creation date	1999	2017	2001	2008	2009	2007	2014
Development status (2019)	stalled	early	stalled	mature/ early for web	active	mature	active
Software License	GPL3	Apache 2.0	Common Public License 1.0	MIT License	GNU LGPL v2.1	BSD-2 Clause	Copyright © 2001 UC Regents
PUBLIC WEB INSTANCE		●		●			●
Creation date		2017		2018			2000
Development status (2019)		early		early			active

Native app (native application): the viewer is an independent application

Web app (web application): the viewer is run on a local server by a web browser. Two bullets means that it can be embedded in a website.

Creation date: year of the first release (either for the local version, or for the public web instance)

Development status: *early*: some developments are needed for the tool to be fully operational / *active*: the tool is fully operational and still actively developed with new functionalities / *mature*: the tool is fully operational and well-maintained / *stalled*: there is no major recent updates.

Software license: all tools are open-source at least for personal, academic and non-profit users.

Artemis
https://www.sanger.ac.uk/science/tools/artemis
Artemis is one of the first genome browsers and was developed by the Wellcome Sanger Institute. It was designed for prokaryotic genomes (it predates the sequencing of the human genome). It is a powerful tool to analyse small to medium-sized prokaryotic and eukaryotic genomes.

GIVE - Genomic Interaction Visualization Engine
https://zhong-lab-ucsd.github.io/GIVE_homepage/
GIVE is a recent viewer, developed by Zhong Lab (University of California San Diego). It was initially specifically designed to visualise genomic interactions (hence its name). But now GIVE's utilities are outgrowing its original purpose to accept other data types.

IGB - Integrated Genome Browser
https://bioviz.org/
IGB is one of the early genome browsers. It was first developed at Affymetrix to support visual analytics of genome tiling arrays. It was then released as an open source software in 2004. Since 2008, IGB has been mainly maintained and developed at the University of North Carolina at Charlotte, with contributions from the community. This is the official viewer of the TAIR project (*Arabidopsis thaliana*), for instance.

IGV - Integrative Genomics Viewer
https://software.broadinstitute.org/software/igv/
IGV is one of the most popular genome viewers, developed by the Broad Institute. It supports a wide variety of data types including NGS alignments, genomic annotations, expression data, genetic variations, etc. This makes it a reference tool. Since 2018, IGV also exists as a web-based viewer.

JBrowse
https://jbrowse.org/
JBrowse pioneered the world of modern web-based genome browser with a client-server architecture, based on JavaScript, CSS and HTML5. It is a GMOD project with major contributions from the Evolutionary Software Foundation. The web application is also available as a stand-alone application through an Electron wrapper (*Jbrowse desktop*).

Tablet
https://ics.hutton.ac.uk/tablet/
Tablet is a lightweight viewer, specifically optimized to deal with assemblies and alignments. It is developed at the James Hutton Institute.

UCSC Genome browser - University of California Santa Cruz
http://genome.ucsc.edu/index.html
The initial prototype of the UCSC genome browser has been released at the same time as the first human assembly in 2000. At the time, it was available only as a public web instance (which was new). It has been continually developed to include a broad collection of eukaryotic organisms as well as new data sources and formats. Since 2014, the USCS genome browser also exists as a local application, with great sharing facilities.

2. Methodology, datasets

We installed all local viewers (both native app and web app when necessary) under Linux. Web-based browsers (GIVE, IGV, Jbrowse and UCSC) were all tested with Firefox. GIVE and UCSC were also tested with Chrome. The version taken into consideration for this study was the latest available release in date of May or June 2019.

We created **two main datasets** with multiple tracks and formats to evaluate all tools on a common basis. For properties that could not be tested in this setting, we relied on documentation provided with the tools. Dataset 1 also served to RAM measurements presented in Section 7 RAM and time requirements.

Evaluated version

Artemis	GIVE	IGB	IGV	Jbrowse	Tablet	UCSC
V18.0.2 (Feb 2019)	Local install: v0.2.0 (Feb 2018) Public web: v0.2.0 (Feb 2018)	v9.0.2 (Nov 2018)	Local install: IGV desktop 2.5.2 (Apr 2019) Public web: v2.2.11 (Jun 2019)	v1.16.5 (Jun. 2019)	1.19.05.28 (May 2019)	Local install: v380 (Apr 2019) Public web: not relevant

Dataset 1: RNA-seq hg38

The first dataset is composed of two samples from the RNA-seq experiment PRJEB8960 (blood biomarkers of risk for synucleinopathy, a variant of parkinson's disease) available on SRA:

- SAMEA3312229 (R1) and SAMEA3312230 (R2): 52,159,829 spots (104M paired-end Illumina reads)
- SAMEA3312235 (R1) and SAMEA3312236 (R2): 24,808,528 spots (50M paired-end Illumina reads)

FASTA, GFF3 and GTF files for hg38 were downloaded from Ensembl (GRCh38 release-82). The two read samples were mapped separately on the genome with STAR (version: 2.6.0a). This generated two BAM files:

- Sample01.BAM for SAMEA3312229 and SAMEA3312230
- Sample02.BAM for SAMEA3312235 and SAMEA3312236

We also used two VCF files:

- Chr20_phase3.vcf, retrieved from 1000genomes
- Chr20_phase3_ligth.vcf, where only one random individual was extracted from the 1000genomes file

At the end, this gives the following tracks.

File names	Size	File format	Content	Source
Genome.fa	3GB	FASTA	Reference genome sequence	Ensembl
Genome.gtf	1.5GB	GTF (GFF2.2)	Reference genome annotation	Ensembl
Genome.gff3	388MB	GFF3	Reference genome annotation	Ensembl
Genome.gff2	433MB	GFF2	Reference genome annotation	Bioconvert (from GFF3)

Sample01.BAM Sample02.BAM	11GB 3GB	BAM	Alignments	STAR
Sample02.BED	2.8GB	BED	Alignements	BAM2BED (from BAM)
Chr20_phase3_ligth.vcf Chr20_phase3.vcf	273MB 18GB	VCF	Variants	1000 genomes

Since some tools require file indexes to improve their performances (see section 7. RAM and time requirements), we also generated a FAI file (FASTA index) with samtools faidx, a BAI file (BAM index) with samtools index, and a TBI file(GFF and VCF index) with htslib tabix. Since some tools accept only BED files instead of a BAM files, we also converted BAM files with BAM2BED. Since some tools support only GFF2 format, we converted the GFF3 file with bioconvert (see section 5. Supported types of data).

Dataset 2 : *Synechococcus elongatus*

The second dataset concerns the freshwater cyanobacterium *Synechococcus elongatus* PCC7942.It consists of two complementary datasets from a ChIP-seq experiment (SRA):

- SRR1005033: WT culture, **anti-RpaA**, ZT 44h, replicate 2
- SRR1005035: WT culture, **input DNA**, ZT 44h, replicate 2

Reference files (FASTA and GFF3) were downloaded from NCBI Genome database (https://www.ncbi.nlm.nih.gov/genome/430?genome_assembly_id=300122).

We performed a simplified ChIP-seq analysis using the following protocol:

1. Mapping the reads from each dataset onto the reference genome using Bowtie2 , creating 2 BAM files,
2. Computing and comparing BAM coverage with deepTools, creating 3 BIGWIG files,
3. Peak calling with MACS2, which resulted in 2 BED files and 2 BEDGRAPH files.

As previously, we also generated FAI, BAI, TBI, BED and GFF2 files with samtools faidx, samtools index, tslib tabix, bam2bed and bioconvert respectively.

File names	**File format**	**Content**	**Source**
genome.fa	FASTA	Reference genome sequence	NCBI
genome.gff	GFF3	Reference genome annotation	NCBI
control_bowtie2.sam treat_bowtie2.sam	SAM	Mapped reads against the genome	Bowtie2
control_bowtie2.bam treat_bowtie2.bam	BAM	Binary version of the SAM files	Samtools
control_bowtie2.bw treat_bowtie2.bw	BIGWIG	Aligned reads coverage	deepTools
compare.bw	BIGWIG	Treatment vs. Control coverage comparison	deepTools
control_lambda.bdg	BEDGRAPH	MACS2 CallPeak fragments pileup	MACS2

treat_pileup.bdg			
narrowPeak.bed	BED6+4	Peaks	MACS2
summits.bed	BED	Peaks summits	MACS2

3. Technical features

This section covers all aspects of installation, administration and configuration of the viewers. The underlying questions are : What is the level of computer literacy required to perform all these tasks? Should I need to be an expert to be able to install and use the tool by myself ? (if not, I will ask my favorite bioinformatician). We also indicate the existence of plugins, for those who would like to contribute to the development of new features.

The information on dependencies, OS, ease of installation, track configuration, plugins applies to local install tools only.

	Artemis	**GIVE**	**IGB**	**IGV Native app**	**IGV Web App + public instance**	**JBrowse Web app**	**JBrowse Native app** *(Desktop)*	**Tablet**	**UCSC**
Development language(s)	JAVA	Javascript HTML	JAVA	JAVA	Javascript	Javascript HTML5 Perl	Electron Javascript HTML5	JAVA	C Shell HTML Perl Javascript MySQL
Dependencies	JAVA 9+	docker	none	Java 6	Node ≧ v8.11.4 NPM ≧ v5.6.0	web server	libgconf (linux)	none	docker
OS supported	all	all	all	all	all	all	all	all	all
Ease of installation	★★★	★★★	★★★	★★★	★★★	★★★	★★★	★★★	★★★
Track configuration	★★	★	★★	★★★	★★★	★	★★		★★
Existence of Plugins			●			●	●		

☆☆☆ none, ★☆☆ minimal, ★★☆ good, ★★★ excellent

OS supported: all viewers run under Windows, Mac OS and Linux. We did not test all versions of those OS.

Ease of installation: we did not encounter any problems in the installation of each of these tools.

Track configuration: we evaluated how difficult is was to add a new track. Does one need to perform a pre-processing (command line) before uploading the file ? Is the viewer able to generate the index in place of the user ?

Existence of plugins: those software components allow to add a specific feature to the genome browser. It is therefore a powerful way to customize the tool, because it gives the possibility to design its own plugins or to use plugins developed by the community.

Artemis: the source code is available on GitHub, or via conda. For track configuration, it is recommended to generate index files. As for colour configuration and text label size, users have to modify the *etc/options* text file, which is not very convenient.

GIVE provides two ways for local install: manual install from source code (expert level) or with Docker image (easy). GIVE is supported by all major browsers such as Firefox, Google Chrome. Adding a new track must be done via command lines in the Docker container. Choosing tracks and genome coordinates implies manipulating HTML code, which could be an obstacle. Overall, it seems that this tool was not really designed to be installed locally and used by multiple users.

IGV Web App requires a modern web browser with support for Javascript ECMAScript 2015 (Mozilla Firefox, Google Chrome, Apple Safari, Edge). It does not support Internet Explorer (IE).

IGV Native App: Native installer files are available for all platforms. The source code of IGV is also available in a GitHub repository and as a Bioconda package (https://anaconda.org/bioconda/igv).

IGB: Native installer files are available for Linux, MacOS and Windows. IGB source code is also available in a SourceForge repository.

JBrowse Web App: The installation is pretty straightforward: download the archive, make it accessible to your HTTP server, then run the install script. Supported web browsers: Mozilla Firefox (10 and later), Google Chrome (17 and later), Apple Safari (9 and later), Microsoft Internet Explorer (11 and later).

JBrowse Native App (Desktop): Native installers are provided for Linux, MacOS and Windows. JBrowse Desktop runs inside an Electron app and does not require a web browser.

Tablet: An installer is provided for Windows (64 bit), Linux (64 bit) & macOS. Alternatively you can use Bioconda.

UCSC provides four ways for local install: manual install from source code (expert level), automated install with a provided script, install in Docker with provided Dockerfile (easy), install in a virtual machine with provided .vbox file (easy). Supported browsers, for both local install and web install : an up-to-date browser that supports JavaScript with *cookies enabled*, such as Firefox 3.0 (and higher), Internet Explorer 6.0 (and higher), Safari 3.0 (and higher). The local install is a complete mirror of the public web instance. By default, after the install no data is included and all the needed data will be downloaded on-the-fly (which requires to always have an Internet connection). But the installer can also choose to download a particular assembly and thereby disable the on-the-fly mode.

4. Ease of use

We describe how a new user can appropriate the tool: Is the genome browser easy to learn and use at first try? Is it easy to understand how it works? Does it come with a good documentation?

	Artemis	GIVE	IGB	IGV Native App	IGV Web App + public instance	Jbrowse	Tablet	UCSC
Getting started	★	★	★★	★★★	★★★	★★	★★★	★★
Documentation, assistance								
User guide	★★★	★★	★★★	★★★	★★★	★★★	★★★	★★★
Online tutorials	★★		★★★	★★	★	★★★		★★
Support	★★		★★★	★★	★★	★★★	?	★★
Community	★		★★	★★	★★	★★★		★★
Developer guide	★★	★★	★★★	★★★	★★★	★★★		★★★

☆☆☆ none, ★☆☆ minimal, ★★☆ good, ★★★ excellent

Getting started: is the viewer easy-to-use and intuitive ? It is possible to achieve simple tasks without the need to refer to an external documentation ?

Documentation, assistance:
User guide: conventional documentation
Online tutorials: walk-throughs, video tutorials, etc.
Support: help supplied directly by the authors or the developers of the viewer, through helpdesk or issues on GitHub for example.
Community: existence of a user's community, active on forums for example
Developer guide: documentation specifically designed for developers (plugins, config files,...)

Artemis
Getting started: It is not so easy. Loading files with the file manager should not be complicated if the indexed files already exist or if you access them via EMBL file or an ftp server. Artemis is composed of several panels for containing different types of data. However, navigating between these different panels is not easy at first try. We regret the lack of a search bar on the interface for example to go to a specific position interval or without having to go through the navigator of the GoTo menu to search for features. Displacements are done via a horizontal bar scroll and zoom via a vertical bar scroll, the panels do not necessarily synchronize their zoom level.
User guide : manual (https://sanger-pathogens.github.io/Artemis/Artemis/artemis-manual.html)
Online tutorials : some videos on YouTube made by community
Support : an active GitHub (https://github.com/sanger-pathogens/Artemis/) or mail contact
Developer guide: mixed inside the user manual

GIVE
Getting started: Although the first steps with GIVE are pretty well documented, it is not really intuitive: the user must use the instance Data Hub to choose the tracks, then the HTML generator will create a piece of HTML

code that the user has to save and open in her/his web browser to access the GIVE genome browser with the desired tracks.
Documentation: it is reduced to a user and a developer guide which are both minimal. All information is gathered in a GitHub repository, but the level of activity seems to decrease.

IGB uses a traditional java desktop GUI that will feel familiar to new users, but this interface is a bit crowded by default which means that it is not always easy to find even basic parameters. However, IGB provides a very exhaustive documentation for both users and developers. Online tutorials are available in a Youtube channel. Support is also provided through multiple ways (issue tracking site, feedback form, email, BioStars forum).

IGV Native App
Getting started is easy. IGV is made to facilitate the use of visualisation of data. For example, it provides build-in functions to index files. The information is It allows you to easily have the information you want to visualise, and can be simply customized. Some parameters can be difficult to understand at first try, but this is well documented in the user guide.
User guide: https://software.broadinstitute.org/software/igv/UserGuide)
Online tutorials: videos on YouTube (https://www.youtube.com/channel/UCb5W5WqauDOwubZHb-IA_rA)
Support: GitHub (https://github.com/igvteam/igv)
Community: Google groups (https://groups.google.com/forum/#!forum/igv-help)
Developer Guide: in UserGuide

IGV Web App
Getting started: this version is very much inspired by the original IGV Native App. So there are few specific documentation. Like the web app integrates less features than the native app, it is even easier to use.
User guide: UserGuide (https://igvteam.github.io/igv-webapp/)
Online tutorials : GitHub (https://github.com/igvteam/igv-webapp)
Support: GitHub (https://github.com/igvteam/igv-webapp)
Community: Google groups (https://groups.google.com/forum/#!forum/igv-help)
Developer Guide: GitHub (https://github.com/igvteam/igv.js/wiki)

Jbrowse offers a lot of documentation and it is easy to achieve a basic visualisation of your data. On the other hand, even if the documentation is great, achieving a custom configuration can be tricky and will need a certain amount of time. Support/community : multiple contacts and supports (GitHub, mailing list …)

Tablet
Getting started is very easy. It's a real strength of the tool.
User guide: http://tablet.hutton.ac.uk/en/latest/
FAQ: https://ics.hutton.ac.uk/tablet/tablet-faq/
There are no other sources of documentation, but this is not a problem because the tool is easy to use with intuitive interface. We never felt in trouble. One open question: we did not find any developer guide (while Tablet is open source).

UCSC

Getting started: Starting with UCSC using public tracks is relatively easy. As soon as you want to add personal tracks, one must read the documentation to load a file correctly and potentially have already stored its data on a remote server or hub.
Documentation: UCSC provides a lot of documentation, maybe too much. It is not always easy to find what you are looking for, or to find information that you have seen previously in the documentation. Online tutorials are available in a Youtube channel (18 videos, https://www.youtube.com/channel/UCQnUJepyNOw0p8s2otX4RYQ)
Support: issues on GitHub
Community: Google groups
Developer Guide: yes (with a conventional documentation and a Wiki).

5. Supported types of data

Reference genome and annotation

	LOCAL INSTALL VIEWERS						
	Artemis	**GIVE**	**IGB**	**IGV app**	**Jbrowse**	**Tablet**	**UCSC**
Supplied genomes	none	hg19	43 species	155	none	none	on-the-fly download from public instance
Custom genomes	GenBank EMBL GFF FASTA (or fai)	UCSC cytoBandIdo	FASTA 2bit BNIB MPFA	FASTA GTF GFF broadinstit ute	FASTA TWOBIT GFF	FASTA FASTQ	TWOBIT GTF GFF2

	PUBLIC WEB INSTANCES		
	GIVE	**IGV web**	**UCSC**
Supplied genomes	mm9, mm10, hg19, hg38	16 assemblies (13 species)	~200 assemblies (~100 species)
Custom genomes	UCSC cytoBandIdeo	FASTA	TWOBIT GFF

Supplied genomes: genomes that are provided with the viewer, by default.

Custom genomes: genomes added by the user. For those genomes, we list the supported formats : FASTA, FASTQ, EMBL or twoBit to store genomic sequences, GenBank, GFF (and variants GFF2, GTF) for genes and other features.

UCSC: The addition of custom genomes is done with Assembly hubs.

Other data: reads, variants, ...

This is all information related to features, sequencing data, quantitative analysis, variant calling... We list below the supported formats. In section 6. Navigation, visualisation, we describe how those formats are rendered at display by the viewers. In section 7. RAM and time requirements, we give some measures for RAM consumption and discuss the sizes of the files that can effectively be loaded by the tools.

Type of track	**Type of biological information**	**File format**	**Artemis**	**GIVE**	**IGB**	**IGV**	**IGV web app**	**Jbrowse**	**Tablet**	**UCSC**
Features	genome annotation, genes, alternatives transcripts, ...	GFF	●		●	●			●	●
		GTF (GFF2.2)			●	●				●
		GFF3	●		●	●	●	●	●	
		BED		●	●	●	●	●	●	●
		BIGBED (binary)			●	●	●	●		●
		GenBank	●		○			●		
Reads alignments	DNA-seq RNA-seq miR-seq ChIP-seq BS-seq ...	SAM			●	●	●		●	
		BAM (binary)	●	○	●	●	●	●	●	●
Quantification	RNA-seq ChIP-seq expression array	BEDGRAPH			●	●	●			●
		BIGWIG (binary)		●	●	●	●	●		●
		GCT/RES (gene expression)				●				
Variant calling	SNP array, DNA-seq, RNA-seq	VCF	●		●	●	●	●		●
		SNP				●				
Long-range interaction	chromatin interaction	Hi-C		●						○
		Others relevant formats	Wig CRAM	UCSC known genes	Cytoband	29	TDF BedPe		AC AFG MAQ	

Blank cell: the format is not supported, ○ possible with some limitations, ● no problem

GIVE

- *BAM file:* the alignment files should be converted to BED files with BAM2BED.

IGB

- *BAM:* BAI index file needs to be available

UCSC

- *VCF:* large VCF files need to be bgzipped and indexed with tabix.

- *Chromatin interactions*: the viewer accept interaction files as bigInteract track format. It displays pairwise interactions as arcs of half-rectangles connecting two genomic regions on the same chromosome.

Public resources

Access to external resources, such as public databases or annotation tools, is a useful functionality which is of great help for gene prediction and analysis. For each viewer, we list the databases and tools that can be queried directly. We also provide a general evaluation of the practicality of these functionalities (third line).

Artemis	GIVE	IGB	IGV	Jbrowse	Tablet	UCSC
EBI Interpro PFAM RFAM Blast	ENCODE	UCSC RefSeq IBG Quickload DBs Blast	ENCODE 1000 genomes TCGA Ensembl genes UCSC BLAT	UCSC Bio::DB	None	UCSC Ensembl 1000 genomes ENCODE PFAM OMIM ClinVar RepeatMasker BLAT
★★★	★★	★★★	★★★	★		★★★

☆☆☆ none, ★☆☆ minimal, ★★☆ good, ★★★ excellent

Artemis: It is possible to load data with an accession number from EBI with Dbfetch. Accessing data on a remote server with ssh is an alternative possibility.

GIVE : Existence of an instance ENCODE (given as an example on GIVE website). A few dozen of ENCODE tracks are available on the public GIVE Data Hub, but there is no direct connection to ENCODE database.

IGB: It is possible to set up a *IGB QuickLoad site* in order for IGB users to download data. The main IGB QuickLoad site already makes available public datasets in RNA-seq, DNA-seq, ChIP-seq, ...

IGV : load track from for Human Hg19:

- ENCODE (tracks from UCSC, DNase clusters, broad histone, regulatory transcription..)
- 1000 Genomes
- TCGA (The Cancer Genome Atlas)
- Ensembl Genes
- Others (cBioPortal, OMIM, GTEx portal, MGC Genes, RGD Human QTL, Platinum Genomes...)

Jbrowse: If you have an existing database with a compliant scheme (such as Chado), then you can use JBrowse's script biodb-to-json.pl to produce several tracks. If you have a local dump of the UCSC genome annotation MySQL database, then you can use JBrowse's script ucsc-to-json.pl.

Tablet: it does not provide any connection to external resources, which is consistent with the scope of this tool.

UCSC: You can use the "track search" functionality or scroll below the viewer to see the available resources.

Tracks are classified into the following categories: Mapping and Sequencing, Genes and Gene Predictions, Phenotype and Literature, mRNA and EST, Expression, Regulation, Comparative Genomics, Variation, Repeats. Among the available resources: UCSC, Ensembl, 1000 genomes, ENCODE, PFAM, OMIM, ClinVar, RepeatMasker… The following tools are directly accessible from the genome browser: Blat, Table Browser, Variant Annotation Integrator, Data Integrator, Gene Interactions, Gene Sorter, Genome Graphs, In-Silico PCR, LiftOver, VisiGene, DNA Duster, Protein Duster, Phylogenetic Tree PNG Maker.

6. Navigation, visualisation

This section reports on the navigation experience: how easy is it to mine the data, to locate regions of interest, to visualise SNPs or quantitative information, to move from the gene scale to the genome scale, to handle multiple tracks.

		Artemis	**GIVE**	**IGB**	**IGV native app**	**IGV web app + public**	**Jbrowse**	**Tablet**	**UCSC**
Search	By annotation	●	●	●	●	●	○	○	●
	By genomic location	○	●	●	●	●	●	●	●
	By sequence similarity	●		●	●		○	○	●
Zoom	Click on a button	○		●	●	●	●	●	●
	Scroll		○	○					
	Click and drag on a zone	○		●	●	●	●	●	●
Tracks	Can modify track order		○	●	○	●	●		●
	Can modify track height	●		●	●	●	○		●
	When high number of tracks		Scroll	Scroll	Scroll	Scroll	Scroll		Scroll
	Distinct panels for visualising distinct regions		●	○	●				
	Autoscale			●	●	●	●		●
	Can overlay tracks	○			●				●
	Can add/subtract quantitative tracks	○		●	●		●		
	Heatmap for quantitative tracks	○		●	●		●		
Visual aspect	Reads orientation (+/-)	●	●	●	●	●	●	●	●
	Features display	●	●	●	●	●	●	●	●

	Features orientation	●	●	●	●	●	●	●	●
	Positive/negative values for quantitative data	○			●	●	○		●
	Alternative mRNA transcripts		●	●	●	●	●		●
	Collapse/expand			●	●	●	●		●
	Annotations	●	●	●	●	●	●	○	●
	Colour-blind friendly			○	○	○	○	○	
Metadata display	Existence of tags			○	●	●	●	○	●
	Dynamic and complete display of metadata			●	●	●	●	○	●
Customi sation	Colour choice			●	●	●	●	●	○
Smooth ness		○	○	○	●	●	●	●	●

blank cell: no functionality, ○ possible but tricky or with some limitations, ● possible and easy

Search by sequence similarity is performed either with BLAST (Artemis, IGB) or BLAT (IGV, UCSC). Tablet and Jbrowse allow to query the reference sequence by exact match or with a regular expression. But there is no real similarity search.

Colour-blind friendly: The user can customize the colours as her/his convenience.

Smoothness: Does the whole experience seem smooth: zooming, scrolling, panning, navigation,...

Artemis

- Automatic display of a six-frame translation panel.
- *Search by genomic location*: Artemis is not able to select a full range (e.g chr1:1-1000). It can only goes to a given position (e.g pos 500 in Chr1).
- *Can overlay tracks* : multiple BAM files are displayed in the track.
- *Heatmap for quantitative tracks* : this is rather a heatmap for the BAM coverage.
- *Smoothness*: using the horizontal scroll bar on large chromosomes causes slow data loading and can affect fluidity. It is better to use the GoTo menu.

GIVE

- *Zoom:* it must be done by scrolling when the cursor is on the coordinate bar. It won't work if the cursor is elsewhere in the window.
- *Can modify track order:* The order of tracks is determined by the order in which the tracks have been configured. It can be changed only by editing the HTML file.
- *Smoothness* depends on the server connection and the data displayed. Visualising the data, zooming in/zooming out… uses a lot of CPU and may take a few seconds for each coordinate change.

IGB

- *Distinct panels for visualising distinct regions:* limited to two regions.
- *Existence of tags* for features or aligned reads: limited to one tag selected amongst only a few main choices.
- *Smoothness:* IGB loading strategy (genome/auto/manual) and its high RAM usage with big BAM or VCF files means that manual loading is always needed when working with big datasets. Therefore, users first need to select the exact region they want to display before manually loading the tracks data, which limit usage smoothness.

IGV Web App

- Can display 6 frames translation, like Artemis.

UCSC

- *Colours:* customisation is not allowed for all file formats.

Format visualisation

In section 5. Supported types of data, we listed all supported formats by the viewers. We discuss here the ways these formats are rendered during visualisation, and which fields are displayed or not. We focus on four main formats: GFF, BAM, BED12+ and VCF. Results are shown in the four tables below.

GFF - 9 fields: seqname source feature start end score strand frame attribute	
Artemis	By default: feature, start, end, attribute Additional information on annotation track : strand, displays start and stop codons as vertical bars
GIVE	Not supported
IGB	All fields are displayed either on mouse-over or on selection
IGV	All fields are displayed either on mouse-over or on selection
Jbrowse	By default, feature is displayed as a tag. Clicking on the feature displays all other fields. Mouse-over shows seqname and description
Tablet	By default, a subset of feature track are displayed (genes, transcript...) and this can be expanded by right clicking. The metadata are displayed by overlay with a popup. We could not find the strand information.
UCSC	seqname, feature, start, end, score

SAM/BAM - 11 mandatory fields: qname flag rname pos mapq CIGAR rnext pnext tlen seq qual + any nb of additional fields	
Artemis	rname, pos, mapq, strand + isize
GIVE	Not supported

IGB	All fields (including additionals) are displayed either on mouse-over, or on selection
IGV	rname, pos, mapq, CIGAR, pnext + additionals fields (e.g library, read length, clipping, tag…)
Jbrowse	On click: pos, length, CIGAR, mapq
Tablet	rname, pos, length, CIGAR, strand
UCSC	On click : flag, rname, pos, mapq, CIGAR, seq, qual, additional fields : band, genomic size, tags

BED12+: 3 mandatory fields: chrom start end, 9 optional fields: name score strand thickStart thickEnd itemRgb blockCount blockSizes blockStarts, 2 additional fields: track ID + url	
Artemis	Not supported
GIVE	Data : Location, exon, intron, CDS, strand, ID.
IGB	Only the first 6 fields can be displayed
IGV	3 mandatory fields + some optional fields : name, score, thickStart, thickEnd, strand (arrows), itemRGB (colored track)
Jbrowse	By default, the metadata is displayed by clicking on the feature. The name, score, positions, strand, length, seqid are displayed in a popup (corresponding to the first 8 fields).
Tablet	The metadata is displayed by overlay with a popup. The fields 5 to 9 are displayed under the tag section. Other fields are missing.
UCSC	On click : chr, start, end. Optional fields : name, score, strand

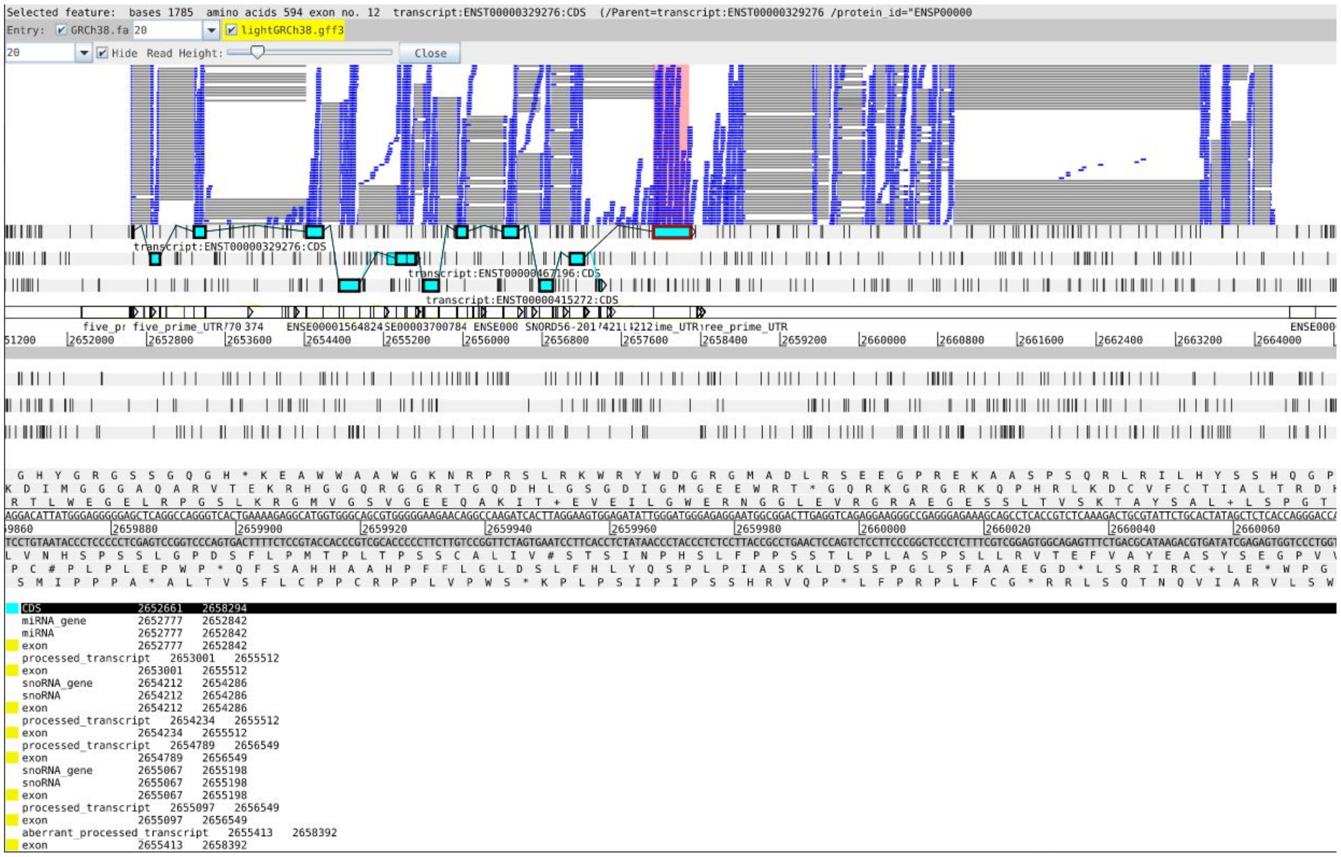
VCF: 8 fixed mandatory fields: chr, pos, id, ref, alt, qual, filter, info, format
INFO=<ID=ID,Number=number,Type=type,Description="description",Source="source",Version="version">
FILTER=<ID=ID,Description="description">
FORMAT=<ID=ID,Number=number,Type=type,Description="description">	

Artemis	pos, id, ref, alt, qual, info (genotype)
GIVE	Not supported
IGB	All fields are displayed either on mouse-over or on selection.
IGV	All fields are displayed either on mouse-over or on selection
Jbrowse	The information displayed depends on the zoom level: either the name and description, or only the name, or even nothing. When clicking on the feature: name, type score, description position, length, attributes (VT, alternative_alleles…), genotypes …
Tablet	Not supported
UCSC	On click : chr, pos, id, ref, alt, qual, filter, info

Look and feel

Deciding which software is the most readable and the most aesthetically pleasing is a subjective question. To give everyone an idea, we present screenshots comparing the 7 viewers. These screenshots were generated on the genomic region Chr20:2,650,000-2,668,000 (18,001 bp) of Hg38, which contains several miRNA and snoRNA loci as well as several mRNAs with multiple alternative transcripts. We added annotations contained in the GFF3 files and alignments contained in Sample02.bam to the visualisation.

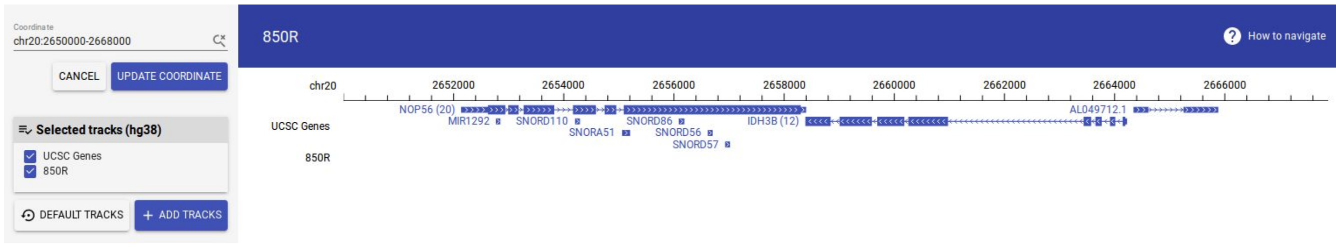
Artemis



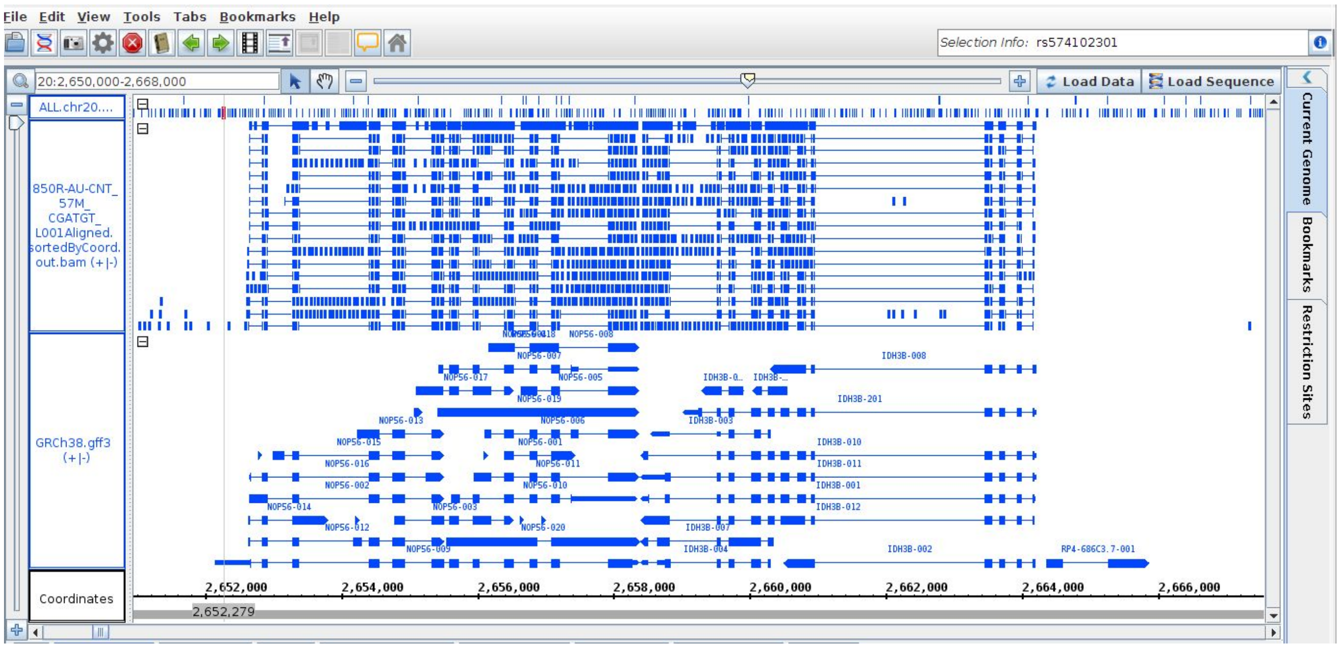
```
Selected feature:  bases 1785  amino acids 594 exon no. 12  transcript:ENST00000329276:CDS  (/Parent=transcript:ENST00000329276 /protein_id="ENSP00000
Entry:  ☑ GRCh38.fa 20  ☑ lightGRCh38.gff3
20  ☑ Hide  Read Height:  Close
```

```
CDS                 2652661   2658294
miRNA_gene          2652777   2652842
miRNA               2652777   2652842
exon                2652777   2652842
processed_transcript   2653001   2655512
exon                2653001   2655512
snoRNA_gene         2654212   2654286
snoRNA              2654212   2654286
exon                2654212   2654286
processed_transcript   2654234   2655512
exon                2654234   2655512
processed_transcript   2654789   2656549
exon                2654789   2656549
snoRNA_gene         2655067   2655198
snoRNA              2655067   2655198
exon                2655067   2655198
processed_transcript   2655097   2656549
exon                2655097   2656549
aberrant_processed_transcript   2655413   2658392
exon                2655413   2658392
```

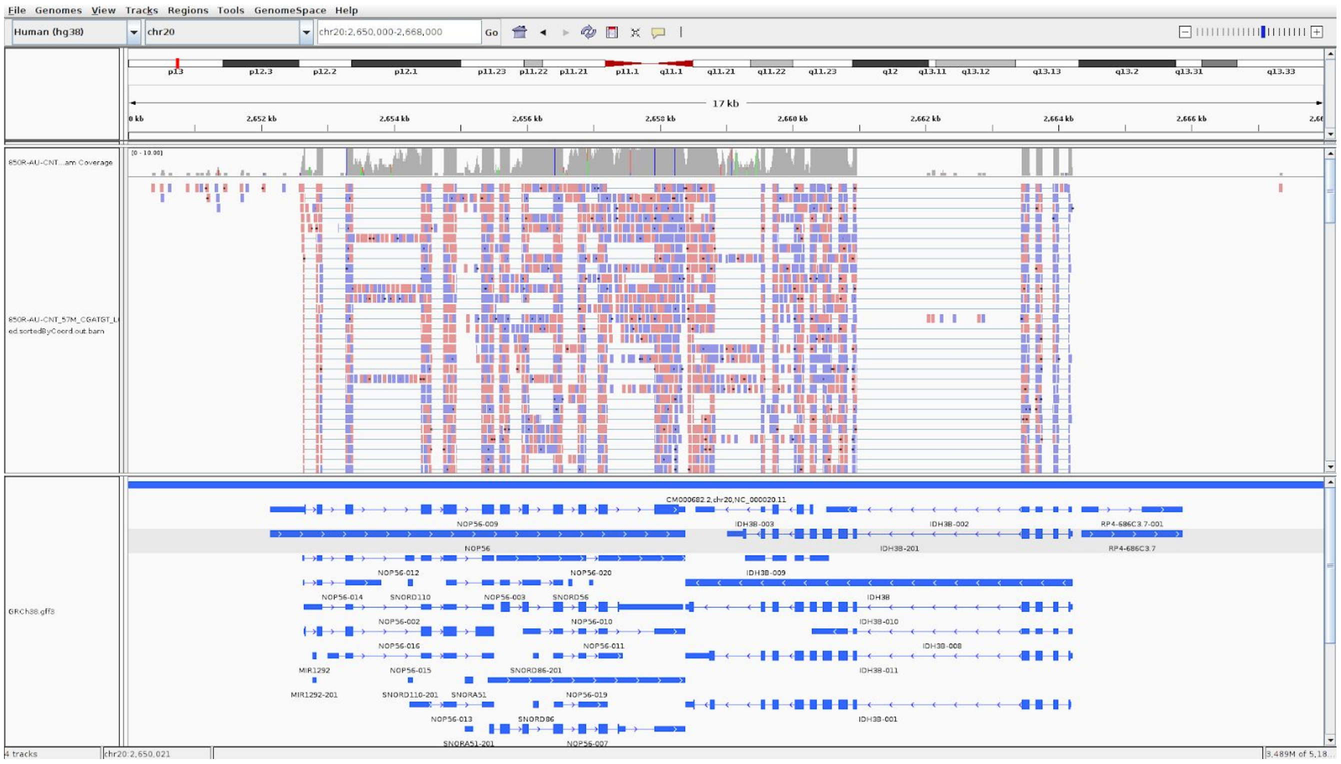
GIVE (GFF3 and BAM are not supported)



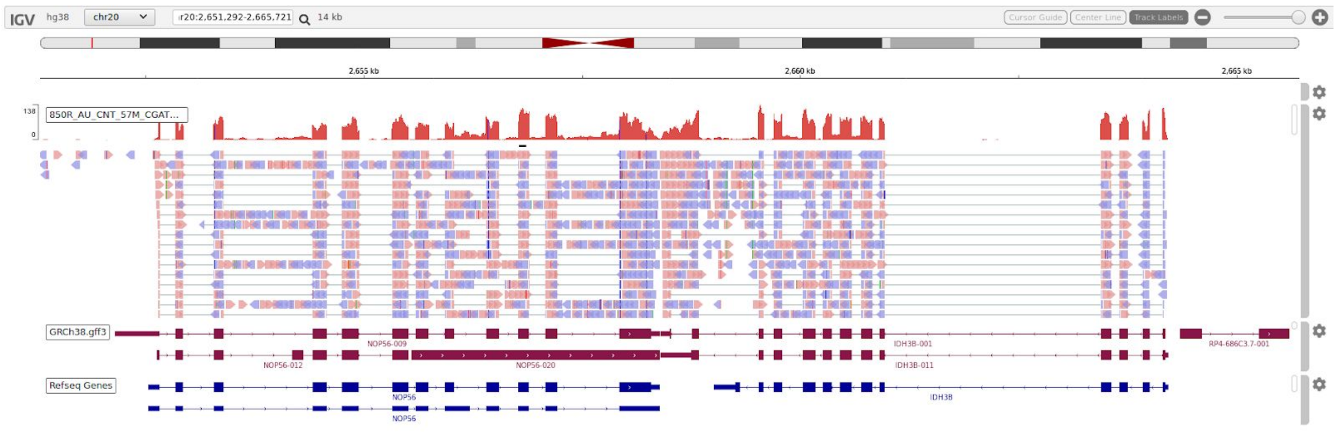
IGB



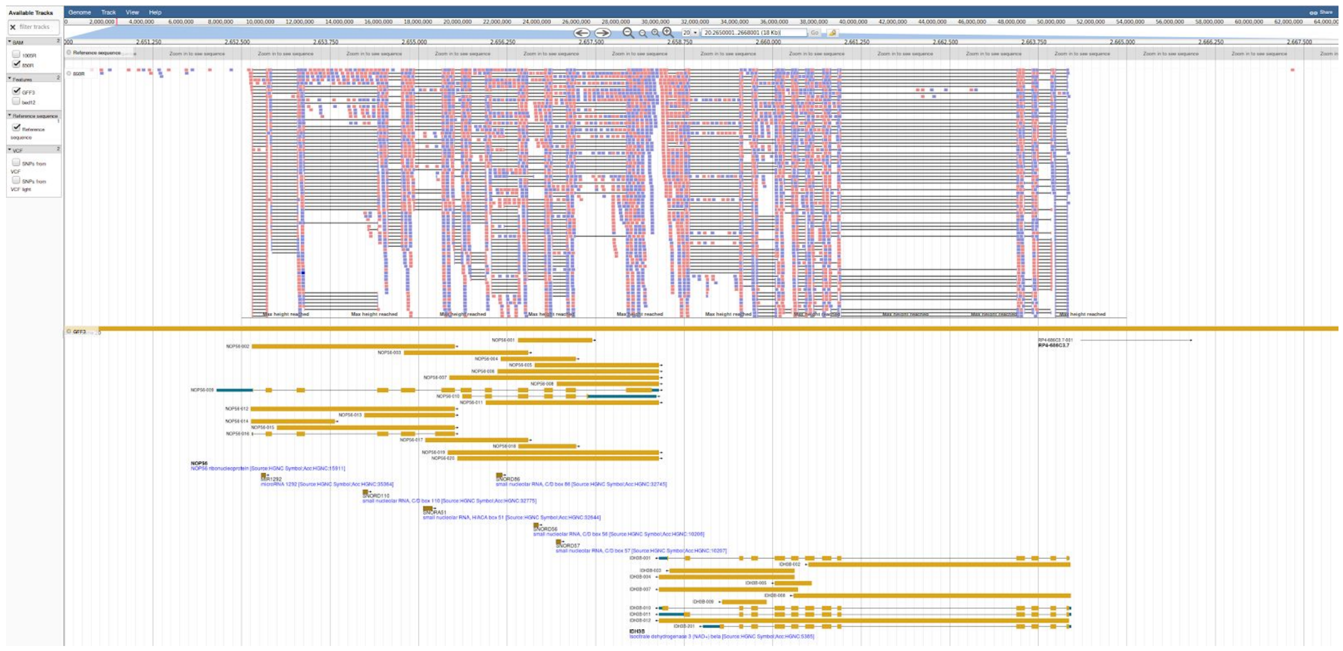
IGV (local install)


File Genomes View Tracks Regions Tools GenomeSpace Help
Human (hg38)
chr20
chr20:2,650,000-2,668,000
Go
17 kb
850R-AU-CNT...am Coverage
850R-AU-CNT_57M_CGATGT_L...ed.sortedByCoord.out.bam
GRCh38.gff3
4 tracks
chr20:2,650,021

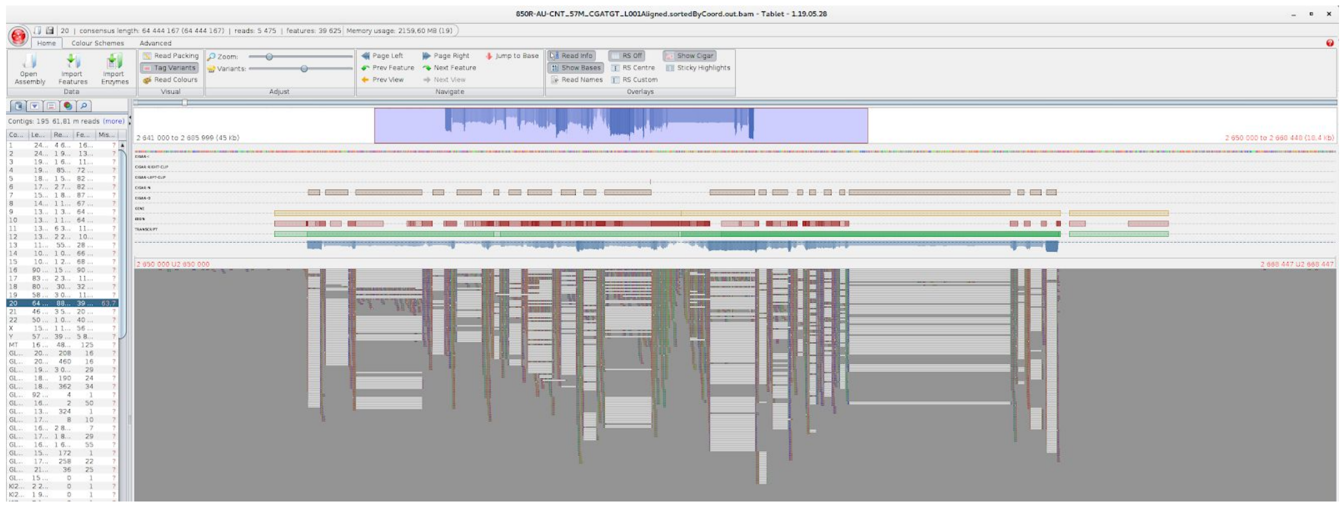
IGV (web application)



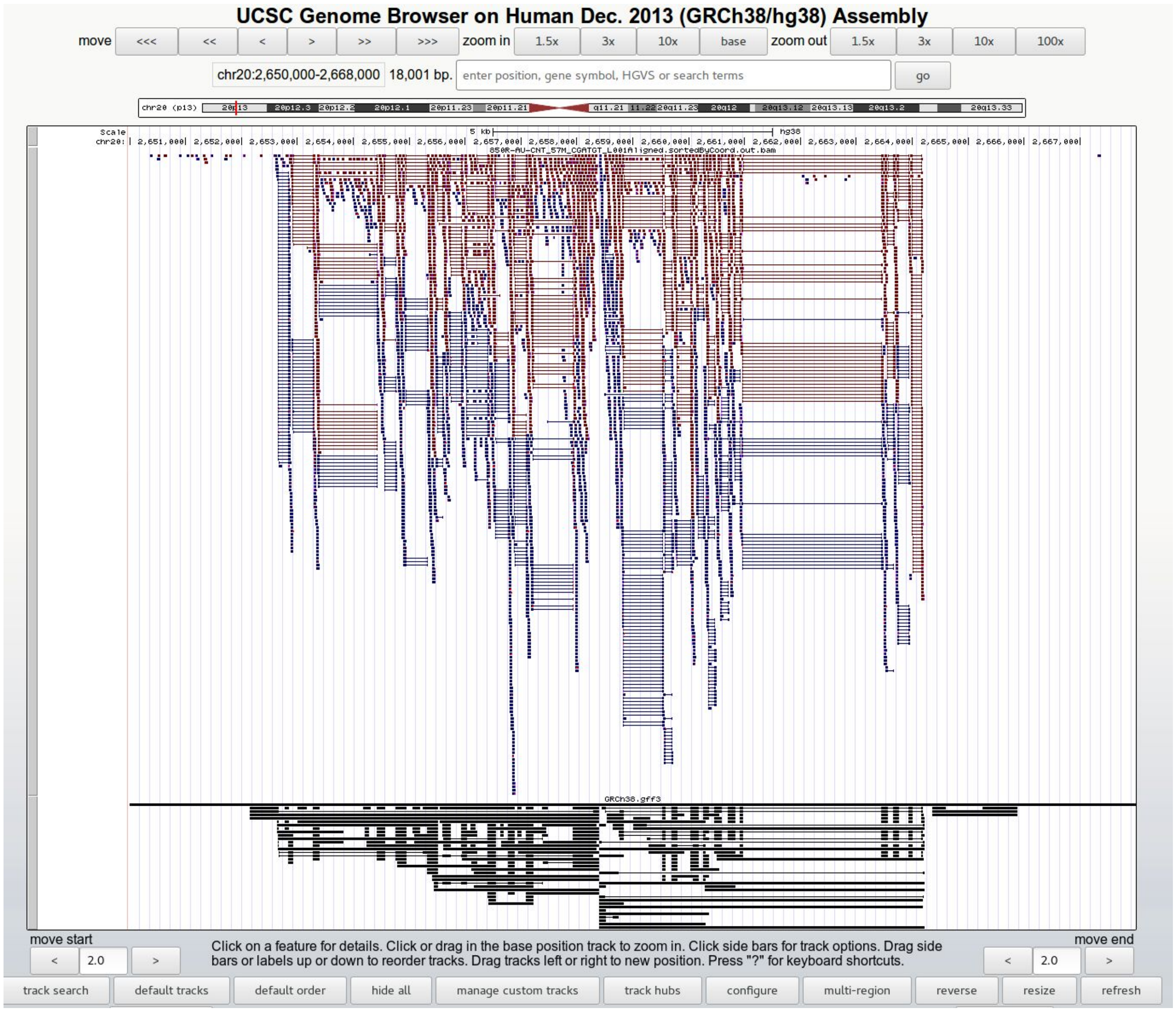
Jbrowse



Tablet



UCSC


UCSC Genome Browser on Human Dec. 2013 (GRCh38/hg38) Assembly
move <<< << < > >> >>>
zoom in 1.5x 3x 10x base
zoom out 1.5x 3x 10x 100x
chr20:2,650,000-2,668,000 18,001 bp.
enter position, gene symbol, HGVS or search terms
go
chr20 (p13)
Scale
5 kb
hg38
chr20:
850R-AU-CNT_57M_CGATGT_L001Aligned.sortedByCoord.out.bam
GRCh38.gff3
move start
move end
track search
default tracks
default order
hide all
manage custom tracks
track hubs
configure
multi-region
reverse
resize
refresh

Click on a feature for details. Click or drag in the base position track to zoom in. Click side bars for track options. Drag side bars or labels up or down to reorder tracks. Drag tracks left or right to new position. Press "?" for keyboard shortcuts.

7. RAM and time requirements

In this section, we have tried to understand the computational limitations of the viewers. For that, we have measured the RAM requirements on a combination of genomic regions of Hg38 and tracks from Dataset 1 (introduced in section 2. Methodology, datasets): we made tests on the longest (Chr1) and the shortest (Chr20) chromosomes with BAM, GFF and VCF files. We also give some indications on the observed loading time.

	Artemis	GIVE	IGB	IGV native app	IGV web app	Jbrowse	Tablet	UCSC
No data (resources needed to launch the viewer)								
RAM	150MB	genome	700MB	genome	genome	100MB	100MB	genome
Hg38 alone (FASTA)								
RAM	150MB	700MB	1.3GB	800MB	100MB	100MB	BAM	500 Mb
Chr20:10,000,000-11,000,000 (1Mb) + Sample02.BAM (3GB)								
RAM	300MB	590MB	1.8GB	500MB	600MB	100MB	350MB	500+MB
Chr20 (64Mb) + Sample02.BAM (3GB)								
RAM Load time			7.4GB 20’’	2.8GB 3-4’		1.3GB 20’’	2GB 20’’	500+MB Few’’
Largest genomic region from Chr1 (249Mb) supported by the viewer + Sample02.BAM (3GB)								
Size (bases) RAM (bytes) Load time	25kb 100Mb 3’	10Mb 900MB 4’	115Mb 12GB 5‘	Full chr 12GB 5’	5Mb 2GB 10’		Full chr 4.3Gb 1’50’’	150Mb 3Gb <10s
Chr1 (249Mb) + GFF (388 MB)								
RAM Time		GFF not supported	3.2GB 5’’	4GB 2’	1.2GB 2’	<100Mb	9GB 1’50’’	3.5GB 10’
Largest genomic region from Chr20 (64Mb) supported by the viewer + VCF (18GB)								
Size (bases) RAM (bytes)		VCF not supported		100kb 4.2GB		5Mb 2GB	VCF not supported	500 Kb 4GB
Largest genomic region from Chr20 (64Mb) supported by the viewer + VCF light (273 MB)								
Size (bases) RAM	Full chr 400Mb	VCF not supported	10Mb 4GB	Full chr 6GB	Full chr 2.3GB	TO DO	VCF not supported	500Mb 4GB

Blank cells mean that the tool did not managed to load the tracks

No data (resources needed to launch the viewer): As the name says, this is the amount of RAM consumed to open the viewer, without loading any data. Note that this test does not apply to GIVE, IGV and UCSC, because they do not run without a reference genome (FASTA or TWOBIT file is needed).

Hg38 only: Tablet could not be tested in this configuration, because it requires a BAM file with the FASTA file.

Chr20:10,000,000-11,000,000 (1Mb) + Sample02.BAM (3GB): we measured the amount of RAM necessary to load a small region (1Mbases) together with Sample02.BAM. For GIVE, we had to use the BED file (2.8GB) converted from the BAM file. For all tools, the time was negligible (few seconds).

Chr20 (64Mb) + Sample02.BAM (3GB):we measured the amount of RAM necessary to load the full chromosome 20 (64,444,167 bp) together with Sample02.BAM. Artemis, GIVE and IGV web app failed in this task. For the other tools, we also report an (approximate) loading time.

Largest genomic region (on Chr1) supported by the viewer + Sample02.BAM: we first tested whether the viewers were able to load the entire Chr1 (248,956,422 bp) together with Sample02.BAM (or Sample02.BED for GIVE). Only IGV native app and Tablet succeeded in this task. For the other tools, we then determined, step by step, what was the size of the longest region from Chr1 that could be loaded by the viewer.

Chr1 + GFF: we tested whether the viewers were able to load the entire Chr 1 together with a GFF file. For UCSC, the format is GFF2 (433MB) and for all other viewers it is GFF3 (380MB). Note that with Tablet the BAM file was loaded too, since this file is mandatory.

Largest genomic region from Chr20 supported by the viewer + VCF (18GB): No viewer managed to load the full VCF file of 18GB on Chr20. So we tried to determine the maximal size of a region of Chr20 for which the VCF file could be loaded.

Largest genomic region from Chr20 supported by the viewer + VCF light (273MB): We then asked the same question with a smaller VCF file.

We also report in the table below if the tools need additional index files, generated by external tools (see section 2. Methodology, datasets)

	Artemis	**GIVE**	**IGB**	**IGV**	**Jbrowse**	**Tablet**	**UCSC**
FAI	●				●		
BAI	●	non relevant	●		●	●	●
TBI GFF	●	non relevant					
TBI VCF	●	non relevant			●	non relevant	●

FAI: fasta index.
BAI: BAM index. Non relevant for GIVE since this tool does not support BAM files.
TBI GFF: GFF index. Non relevant for GIVE since this tool does not support GFF files.
TBI VCF: VCF index. Non relevant for GIVE and Tablet, since these tools do not support VCF files.

Artemis : our tests clearly show that Artemis is optimized for prokaryotic genomes. It is not designed to deal with human genome. It was not able to load the BAM file or the large VCF file with Chr20 (not to mention Chr1). So we could not even try to determine the size of longest region of Chr1 that could be loaded together with the BAM file. With Chr1 and GFF3 files, Artemis cannot load more annotations after 5 Mbases and takes 1GB RAM (~30 seconds). For all these tests, we could not change the settings. For the small region Chr20:10,000,000-11,000,000 with the BAM file, Artemis provided only a cover track rather than displaying the reads. The limit is 25k bp. It means that Artemis optimizes the RAM usage to try to prevent from crashing. In general, providing indexes for FASTA, GFF and VCF files significantly improves the performance of the tool (even if this is not mandatory). With our BAM and GFF files, Artemis really needs the index files, otherwise it cannot load it without crashing. Artemis passed the test with a light VCF file, provided that the VCF file is gzipped and indexed with tabix beforehand. For smaller genomes and smaller BAM and GFF files (such as provided in Dataset 2), we have not encountered the same problems. We observed that 2-4 GB of RAM is sufficient. It is in accordance with the philosophy of Artemis .

GIVE uses typically about 500 MB of RAM to load and display initial data, and 500 MB for each additional track. Overall, we found that the memory management is quite poor, with high RAM consumption. Adding a track may take several minutes.

IGB allows for automatic or manual data loading strategies. For all tests including the BAM file or a VCF file, we had to force loading manually. When we tried to first load the entire Chr1 with Sample02.BAM, it consumed more than 16GB RAM, and then crashed. With the large VCF files, just opening the file (without loading anything) took more than 15 minutes. With the VCF light, we only managed to load a portion of ~10Mb on the Chr20, using 4GB RAM. It took more than 5 minutes. So caution is needed when loading large datasets because of the risk of out-of-memory crashes. IGB is not designed for visualisation on large regions of large genomes. Especially, BAM and VCF visualisation has to be limited to small regions (a few genes). If not, it would require an unreasonable amount of RAM (e.g. >>16GB). We also looked at the CPU usage. Adding tracks doesn't need a lot of CPU resources because of IGB auto/manual data loading strategy. Loading a lot of data at a time can take a maximum of a few minutes at 100% CPU to fill up to 16GB RAM. Overall CPU usage seems under control and was not a bottleneck during our tests.

IGV Native App allows to adjust parameters to regulate RAM consumption. It can parameter the visibility range threshold (kb) and the number of reads per window. This parameter optimizes the RAM and allows to load simultaneously large datasets. Even if the RAM is entirely used, IGV does not crash and displays the loaded data. CPU is solicited to load data or to display a new interval after moving. However, this resource is not restrictive for the smoothness of the navigation. We also noticed that in all tests, IGV alternates between all CPUs (four) and one CPU. Another point worth mentioning: IGV uses index files, like the other viewers. The difference is that it provides everything you need to generate them without any external tool, which is very convenient. IGV can create FAI automatically. As for BAM and VCF files, the user might run the *igvtools*, which is easiest than installing samtools or tabix.

JBrowse Web App and Native app: Since Jbrowse desktop (native app) is an encapsulation of Jbrowse Web app, the RAM usage and the execution time are essentially the same. By default, JBrowse uses a controlled amount of RAM and implements an adaptive strategy for visualisation. It uses density graphs when the number of alignments or annotations to display is too large (this is what happened with the BAM and GFF files), or even provides no rendering and you have to zoom in to access the information (this is what happened with the VCF files). In all those cases, the viewer does not crash and the navigation is not interrupted. There is no RAM overhead, and consequently the measures reported in the Table should be interpreted with caution, with this adaptive strategy in mind. There is also an automatic warning message when an index file is required, depending on the size of the data. If you want to visualise more data, you have to update the configuration by increasing the number of features to display but at the cost of losing smoothness and using more RAM for the client side. This is what we did to load Chr20 + Sample02.BAM, or to try to load Chr1 (which failed).

Tablet: By default, Tablet visualises a window of 25KBases. It takes about 1.5 GB to display this region with the Sample02.BAM file and about 6Go with the GFF3 file, whereas this latter file is significantly smaller. This shows that the memory usage is really well optimized for BAM files compared to GFF files. The GFF files are not so well managed and in this case you can be forced to allow more memory to the Java Runtime machine.

UCSC: It seems that even if the computer has more RAM, the browser will not use more than that amount of 4GB RAM. The loading time is quite quick. If the genome browser cannot do the job, the user will know it in 10 seconds.

8. Sharing facilities, interoperability

In this section, we studied the ability of viewers to export the results in order to make further use of information and share it with other people.

	Artemis	GIVE	IGB	IGV native app	IGV web app + public	Jbrowse	Tablet	UCSC
Graphic export	BMP tiff Tif PNG SVG JPG		PNG JPG SVG	SVG PNG JPG	SVG		PNG	PNG PDF PS
Local export		○	?		○	○		●
Global export	○	○	○	○	○	●		●

Blank cell: no service, ○ some possibilities, ● fully operational, ? we did not succeed in testing it

Graphic export: the viewer allows to create its own screenshots (in PNG or JPG, for example), possibly with customization options.

Local export: exchanges are allowed within a given instance of the viewer (for local install tools).

Global export: exchanges are allowed with anyone you want, either with specific people or publicly.

Artemis does not have any advanced solution for interoperability.

- *Graphic export* allows to print out the contents of the current window. All or some of the window panels can be selected for printing to an image file. Images can be converted to a raster image (e.g. png, tiff) at any resolution by exporting it from other applications. Therefore, the SVG format can be useful for creating publication quality figures.
- *Local export*: one can create a kind of session that works like bookmarks, to load custom files directly from the computer/external hard drive or from a url, but it is not an export, it's look like a local session.
- *Global export* is possible by saving annotation files (created directly in ARTEMIS) in EMBL format and sharing them. Very powerful for sharing annotations in GenBank format.

GIVE has two distinct modalities for local and global exports.

- *Graphic export:* none

- *Local export:* GIVE is accessible through a simple HTML page. So you can share or export your session by simply sharing this HTML page to anyone who can access the same GIVE instance. But the tracks and genomes must be added manually to the docker instance, which requires some technical skills. The user cannot add a new track (there is a button for that but it does not seem to work), she/he can only choose between the already available tracks in the instance.
- *Global export*: there is a Google form to fill in to publish a track on the public GIVE Data Hub.

IGB provides only limited sharing functionalities.

- *Graphic export:* PNG, JPEG or SVG either including or not the surrounding GUI.
- *Local/Global export*: Sessions can be saved to XML format, but datasets files are not bundled together with the session file. In our tests, we couldn't even load session files generated on the same computer, with the same IGB instance.

JBrowse is well-suited for local/global export

- *Graphic export:* none
- *Local export:* you can share a link of your JBrowse instance
- *Global export:* you can export each track as a file. In this latter case, the format depends on the nature of the track: FASTA, BED, BEDGRAPH, GFF3, WIGGLE,...Moreover, you can choose to export the visible region or the whole region. Ultimately, you can copy the directory which hosts your data and configuration to another location.

IGV Native App

- *Graphic export:* SVG, PNG or JPG
- *Local export / Global export:* the working session can be saved in an XML file to be shared. The XML file contains all configurations and paths for tracks. It is then necessary to provide it with the data corresponding to the displayed track for global export.

IGV Web App and Public Web Instance

- *Graphic export*: SVG
- *Local export* (web app): it's convenient to share the working session from an HTTP link, a QR code, an email or EMBED. The only requirement is that the track displays must be accessible from a URL link.
- *Global export* (public web instance or web app with a public server) : same process as above

Tablet has very limited export capacities.

- *Graphic export:* PNG
- *Local/Global export*: none

UCSC offers a large range of solutions.

- *Graphic export*: PNG, PDF or PS. The user can configure the text size, image width, label size, scale...
- *Local export*: a user can freely create an account on the public instance as well as on a local instance. This allows to save sessions, share them (to users of the same instance only) or keep them private. It is also possible to export a plain text file with the session settings (for users of the same instance only).
- *Global export*: Tracks can be shared with the Tracks Hubs from one instance to another (the data are hosted on an external server and not in the genome browser instance). Tracks Hubs can be made public on Public Hubs.

9. A few words of conclusion

Our work shows that there is a great diversity of genome browsers, with different implementation, visualisation, source data and sharing choices. And also different computational performances. Each tool brings its own small stone to the edifice of genome visualisation (some more than others).

Before detailing each viewer, we would like two stress two general observations. First, there is a growing trend of web-based browsers (either local or public), that take advantage of advances in web technologies (JavaScript, HTML5, CSS, responsivity). Second, we regret that no tool offers real possibilities for real-time collaborative work allowing multiple people to work simultaneously on the same project

Artemis is doing well with small microbial genomes. In this context, it offers many valuable features: edition of annotations (correct/add/delete) and connection to PFAM, RFAM..., access to the 6 reading frames, easy export,... With longer genomes however, Artemis loses in efficiency and smoothness. It also has difficulties to manage a higher quantity of tracks. So this viewer cannot be recommended for analysis of eukaryotic genomes. Lastly, even if Artemis is still well-maintained, it is not clear whether new versions with major updates and additional features will be released in the coming years.

GIVE is a new tool that comes with two great promises: supporting interaction data visualisation, and allowing the building of custom genome browser websites with sharing facilities. Based on our assessment, we must however admit that it did not completely convince us. It is difficult to configure news tracks. There are too few accepted formats. Memory management prevents large files from being loaded, etc. Our feeling is that GIVE is based on very good intentions, but it certainly still requires additional developments to reach its full potential.

IGB is a well-recognized viewer, with years of development behind it. It is thus a mature tool that supports a large range of formats and includes many features.However, we have also identified a number of limitations that may make it a less competitive tool today than in the past. The visual interface lacksclarity and sobriety. Some formats are restrictive. Sharing sessions is tricky. Lastly, the memory management induces a high RAM consumption, which could be a problem for large datasets. As a conclusion, if you are not already familiar with IGB, if it is not already installed in your lab, we would not recommend to start a new project with it.

IGV is a very popular genome browser whose success has not been denied over the years. This is well-deserved because it is remarkably effective and versatile. It can work with a large diversity of formats. It is able to load a large number of tracks and to navigate them smoothly. It offers many possibilities for customisation and sharing. And despite this richness, it has remained simple to use. So if you do not know which viewer to choose, IGV is never a bad choice. Our only regret is that the visual appearance of the “historical” local native version is a little bit old fashioned. The release in 2018 of a web-based application is very promising.

Jbrowse is an ambitious project, that has paved the way to modern web-based genome browsers. It still has many strengths, such as a great potential of parameterization, a fine and intelligent memory management that allows a smooth navigation, a nice visual. It also comes with several powerful features that are particularly well suited to bioinformatician developers: connection to Chado databases, plugins. This is all the more true that the tool benefits from excellent documentation. The counterpart is that a good use of Jbrowse sometimes requires quite advanced technical skills. For example, customising tracks is done by editing a config file. So making the choice to work with Jbrowse is not an insignificant choice. This presupposes having access to the right technical skills.

Tablet is defined by its authors as "a lightweight, high-performance graphical viewer for next generation sequence assemblies and alignments". Our evaluation shows that it perfectly fulfils its missions. It is truly easy to install and to use. The visualisation is particularly well-suited for alignments, and memory management is optimised for BAM files. So it has many qualities that makes it a practical and powerful tool. At the same time, Tablet should not be mistaken for a generic genome viewer. For example, it suffers from certain limitations for the support of GFF or VCF files and does not offer any connection to external resources.

UCSC genome browser: Like IGV, UCSC is a reference genome browser that has been able to remain very competitive thanks to continuous improvements over the years. The UCSC web public instance has always been very popular and well updated since its release in 2000. Since 2014, UCSC is also available as a local application, which further broadens its scope. Today, UCSC includes an unparalleled number of recognized formats, genome assemblies and annotation sources for animals. It also offers many attractive functionalities: smooth navigation, intuitive interface, good memory management, local and global exports for sharing sessions (this is the best tool of our evaluation from that point of view). The only disadvantage is that it takes some time to master all its functionalities. It should also be noticed that some file formats require the files to be stored on an external storage server.

10. Quiz

If you wish, you can now test your knowledge of browser genomes: take our quiz.