*Bioinformatics*, 39(1), 2023, btac830
https://doi.org/10.1093/bioinformatics/btac830
Advance Access Publication Date: 23 December 2022
Applications Note

OXFORD

Genome analysis

# igv.js: an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV)

**James T. Robinson[1,*], Helga Thorvaldsdottir[2], Douglass Turner[1] and Jill P. Mesirov[1,3]**

[1]Department of Medicine, University of California San Diego, La Jolla, CA 92093, USA, [2]Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA and [3]Moores Cancer Center, University of California San Diego La Jolla, CA 92037, USA

*To whom correspondence should be addressed.

Associate Editor: Can Alkan

Received on July 19, 2022; revised on November 29, 2022; editorial decision on December 5, 2022; accepted on December 22, 2022

## Abstract

**Summary:** igv.js is an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV). It can be easily dropped into any web page with a single line of code and has no external dependencies. The viewer runs completely in the web browser, with no backend server and no data pre-processing required.

**Availability and implementation:** The igv.js JavaScript component can be installed from NPM at https://www.npmjs.com/package/igv. The source code is available at https://github.com/igvteam/igv.js under the MIT open-source license. IGV-Web, the end-user application built around igv.js, is available at https://igv.org/app. The source code is available at https://github.com/igvteam/igv-webapp under the MIT open-source license.

**Contact:** jrobinso@ucsd.edu

**Supplementary information:** Supplementary information is available at *Bioinformatics* online.

## 1 Introduction

The rapid pace of genomic dataset generation has led to a proliferation of public and private web portals and cloud resources for sharing, analyzing and visualizing the data. An important component of many such resources is the ability to visually inspect the data in track-based views aligned to a reference genome. Traditionally, such views have been provided in genome browsers, including large client–server systems such as the UCSC Genome Browser (Kent *et al.*, 2002) and the Ensembl Genome Browser (Hubbard *et al.*, 2002), rich client web applications such as JBrowse (Buels *et al.*, 2016) and desktop applications such as the Integrative Genomics Viewer (IGV) (Robinson *et al.*, 2011; Thorvaldsdottir *et al.*, 2013). Although it is possible to create links from web portals to visualize data in these and similar browsers, in many cases it is desirable to embed genomic visualizations directly in portal pages, enabling data visualization in place without the context switch to another web page or application. Here, we describe igv.js, an embeddable JavaScript implementation of the IGV genome viewer.

## 2 Features

Unlike stand-alone genome browsers such as JBrowse, the igv.js viewer is a component designed explicitly to be embedded in and controlled by enclosing web applications. It builds on previous work, importantly Dalliance (Down *et al.*, 2011), the first pure JavaScript component to natively support next-generation sequencing (NGS) data in binary BAM files in a web browser. Embedding an igv.js visualization is as simple as creating a container 'div' element in the HTML document and calling a JavaScript function to insert the genome browser (see Quick Start in Supplementary Material). The enclosing application interacts with igv.js by means of an API, which includes functions to configure the initial view, load and remove reference genomes and tracks, navigate to specific loci and listen for events initiated by user interaction with igv.js. Tracks are initialized with configuration objects that define the track type, data source and format and initial visual properties such as color, height and feature layout within the track. Multiple instances of igv.js can exist on a page, and instances can be added and removed dynamically.

The look and feel of igv.js are modeled closely after the IGV desktop application. Like the desktop IGV application, igv.js supports a wide range of genomic track types and file formats, including aligned reads, variants, coverage, signal peaks, annotations, eQTLs, GWAS and copy number variation (see Supplementary Material and examples in Fig. 1). A particular strength of IGV is manual review of genome variants, both single-nucleotide and structural variants (Robinson *et al.*, 2017). To support this use case in igv.js, we have implemented the suite of IGV read alignment display options. These include coloring and sorting reads by strand and other attributes, displaying and coloring bases that mismatch the reference, coloring alignments to flag ambiguous mappings and anomalous pair distance and orientations, and multi-locus views for side-by-side views of the multiple components of a structural variant.

It is important to emphasize that igv.js by design supports industry-standard file formats such as BAM, CRAM, bigWig, bigBed, VCF and others, without modification. No pre-processing is

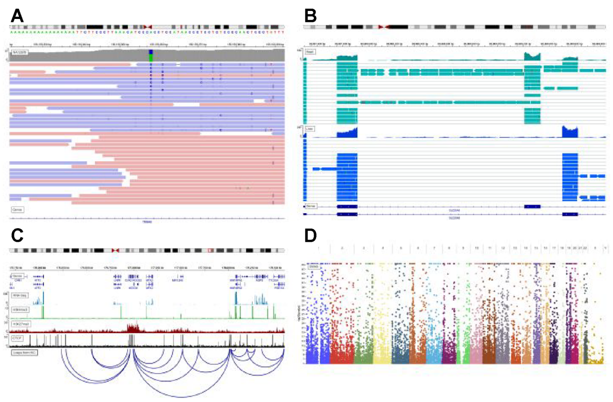

**Fig. 1.** Example igv.js screenshots. (**A**) NGS sequence alignment pileup in the region of a putative SNP indicates strand bias (https://tinyurl.com/y9n6dyw9). (**B**) RNA-Seq pileups from two tissues illustrate alternative splicing (https://tinyurl.com/ybo547uf). (**C**) Epigenetic marks and transcription factors correlate with loop calls from HiC data (https://tinyurl.com/y75465od). (**D**) Manhattan plot from whole-genome association studies (https://tinyurl.com/ydhakvy4)

required. In fact, files in standard repositories, such as those hosted by the UCSC Genome Browser, ENCODE (https://www.encodeproject.org) and GEO (https://www.ncbi.nlm.nih.gov/geo), can be referenced directly by URL. Furthermore, there is no backend IGV server; data are loaded directly from the source into the user's web browser. Supported data sources include (i) files hosted on web servers and accessed via URL; the servers may be on the user's local intranet or on the internet, including on servers hosted by cloud providers such as Google Drive, Google Cloud Storage, Dropbox and Amazon S3; (ii) htsget (https://samtools.github.io/hts-specs/htsget.html) servers; (iii) custom web services; (iv) inline data URIs; and (v) files on the local file system. To support viewing large files, such as reference sequence FASTA, BAM alignment and VCF variant files, indices are used to fetch only the data required to render the current view utilizing standard HTTP Range requests.

Authenticated access to restricted datasets is supported with explicit support for oAuth (https://oauth.net) tokens, which can be supplied by the enclosing application as a static string, a callback function or a JavaScript promise. The support for functions and promises enables the enclosing application to update or revoke tokens as needed. Additionally, arbitrary HTTP request headers can be specified during track initialization to support other token-based authentication schemes. Finally, the URLs to track resources can be specified as a callback function or promise, allowing the enclosing application to intercept and modify the URL at runtime based on user credentials or other criteria. For example, a signed URL might be generated and supplied via a promise after user authentication. Finally, explicit support is provided for Google resources by specifying a Google client id on igv.js initialization. If specified, oAuth authentication will be initiated by igv.js as required.

A unique feature of igv.js is an API function to return a compressed URL-safe string that encodes the complete state of the browser session, including pointers to the reference genome and the loaded data, as well as details on the current view and track and browser settings. This capability can be used by applications to support bookmarks and sharable links. We take advantage of this feature in IGV-Web (https://igv.org/app), an end-user application built around the igv.js component. IGV-Web provides a user interface for specifying the reference genome, loading data into the viewer from remote and local files, and saving and loading sessions. A *Share* menu allows the user to easily retrieve a URL that represents the current session state and to share it with others. The links provided in the legend for Figure 1 were created this way. Entering them into a web browser will open the IGV-Web app and recreate the interactive sessions used to generate the images in the figure.

## 3 Availability and documentation

The igv.js component can be installed from NPM at https://www.npmjs.com/package/igv. The distribution consists of a single JavaScript file with no external dependencies. The source code is available at https://github.com/igvteam/igv.js under a permissive open-source license (MIT) and includes numerous examples for embedding igv.js in web pages. The NPM site includes a quick-start guide, and more extensive documentation is available at https://github.com/igvteam/igv.js/wiki.

The IGV-Web application is available at https://igv.org/app, and the Help menu links out to documentation at https://igvteam.github.io/igv-webapp. The source code is available under the MIT open-source license at https://github.com/igvteam/igv-webapp.

## Funding

This work was supported by the National Institutes of Health [U24CA210004 to J.P.M. and J.T.R., U24CA258406 to J.P.M. and J.T.R.].

*Conflict of Interest*: none declared.

## References

Buels,R. *et al.* (2016) JBrowse: a dynamic web platform for genome visualization and analysis. *Genome Biol.*, **17**, 66.

Down,T.A. *et al.* (2011) Dalliance: interactive genome viewing on the web. *Bioinformatics*, **27**, 889–890.

Hubbard,T. *et al.* (2002) The Ensembl genome database project. *Nucleic Acids Res.*, 30, 38–41.

Kent,W.J. *et al.* (2002) The human genome browser at UCSC. *Genome Res.*, **12**, 996–1006.

Robinson,J.T. *et al.* (2017) Variant review with the integrative genomics viewer. *Cancer Res.*, **77**, e31–e34.

Robinson,J.T. *et al.* (2011) Integrative genomics viewer. *Nat. Biotechnol.*, **29**, 24–26.

Thorvaldsdottir,H. *et al.* (2013) Integrative genomics viewer (IGV): high-performance genomics data visualization and exploration. *Brief. Bioinform.*, **14**, 178–192.