



## COMMENTARY

# BioJS: an open source standard for biological visualisation – its status in 2014 [v1; ref status: indexed, <http://f1000r.es/2yy>]

Manuel Corpas<sup>1</sup>, Rafael Jimenez<sup>2</sup>, Seth J Carbon<sup>3</sup>, Alex García<sup>4</sup>, Leyla Garcia<sup>2</sup>, Tatyana Goldberg<sup>5</sup>, John Gomez<sup>2</sup>, Alexis Kalderimis<sup>6</sup>, Suzanna E Lewis<sup>3</sup>, Ian Mulvany<sup>7</sup>, Aleksandra Pawlik<sup>8</sup>, Francis Rowland<sup>2</sup>, Gustavo Salazar<sup>9</sup>, Fabian Schreiber<sup>2,10</sup>, Ian Sillitoe<sup>11</sup>, William H Spooner<sup>12</sup>, Anil S. Thanki<sup>1</sup>, José M Villaveces<sup>13</sup>, Guy Yachdav<sup>5,14,15</sup>, Henning Hermjakob<sup>2</sup>

<sup>1</sup>The Genome Analysis Centre, Norwich Research Park, Norwich, NR4 7UH, UK

<sup>2</sup>European Bioinformatics Institute EMBL-EBI, Hinxton, CB10 1SD, UK

<sup>3</sup>Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA

<sup>4</sup>School of Library and Information Science, Florida State University, Tallahassee, FL, USA

<sup>5</sup>TUM, Department of Informatics, Bioinformatics & Computational Biology, 5748 Garching/ Munich, Germany

<sup>6</sup>Department of Genetics and Cambridge Systems Biology Centre, Cambridge University, Cambridge, CB2 3EH, UK

<sup>7</sup>eLife, Cambridge, CB2 1JP, UK

<sup>8</sup>Faculty of Mathematics, Computing and Technology, Open University, UK, Milton Keynes, MK7 6AA, UK

<sup>9</sup>Computational Biology Group, University of Cape Town, Cape Town, South Africa

<sup>10</sup>The Wellcome Trust Sanger Institute, Hinxton, Cambridge, CB10 1SD, UK

<sup>11</sup>Biomolecular Structure and Modelling Group Department of Biochemistry, University College London, London, UK

<sup>12</sup>Eagle Genomics Ltd, Cambridge, CB22 3AT, UK

<sup>13</sup>Max Planck Institute of Biochemistry, Am Klopferspitz 18, 82152, Germany

<sup>14</sup>TUM Graduate School of Information Science in Health (GSISH), 85748 Garching/Munich, Germany

<sup>15</sup>Biosof LLC, New York, NY, 10001, USA

**v1** First published: 13 Feb 2014, 3:55 (doi: [10.12688/f1000research.3-55.v1](https://doi.org/10.12688/f1000research.3-55.v1))  
Latest published: 13 Feb 2014, 3:55 (doi: [10.12688/f1000research.3-55.v1](https://doi.org/10.12688/f1000research.3-55.v1))

## Abstract

BioJS is a community-based standard and repository of functional components to represent biological information on the web. The development of BioJS has been prompted by the growing need for bioinformatics visualisation tools to be easily shared, reused and discovered. Its modular architecture makes it easy for users to find a specific functionality without needing to know how it has been built, while components can be extended or created for implementing new functionality. The BioJS community of developers currently provides a range of functionality that is open access and freely available. A registry has been set up that categorises and provides installation instructions and testing facilities at <http://www.ebi.ac.uk/tools/biojs/>. The source code for all components is available for ready use at <https://github.com/biojs/biojs>.

## Article Status Summary

### Referee Responses

Referees	1	2
v1 published 13 Feb 2014	 report	 report

1 **Phil Bourne**, National Institutes of Health (NIH) USA

2 **Reinhard Schneider**, University of Luxemburg USA

### Latest Comments

No Comments Yet



This article is included in the [BioJS Collection](#)

**Corresponding authors:** Manuel Corpas ([mc@manuelcorpas.com](mailto:mc@manuelcorpas.com)), Henning Hermjakob ([hhe@ebi.ac.uk](mailto:hhe@ebi.ac.uk))

**How to cite this article:** Corpas M, Jimenez R, Carbon SJ *et al.* **BioJS: an open source standard for biological visualisation – its status in 2014** [v1; ref status: indexed, <http://f1000r.es/2yy>] *F1000Research* 2014, 3:55 (doi: [10.12688/f1000research.3-55.v1](https://doi.org/10.12688/f1000research.3-55.v1))

**Copyright:** © 2014 Corpas M *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Data associated with the article are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

**Grant information:** We are grateful to the NHLBI Proteomics Center Award (HHSN268201000035C), the UK's Biotechnology and Biological Sciences Research Council (BBSRC) and European Commission grant PSIMEx (FP7-HEALTH-2007-223411).

*The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

**Competing interests:** No competing interests have been disclosed.

**First published:** 13 Feb 2014, 3:55 (doi: [10.12688/f1000research.3-55.v1](https://doi.org/10.12688/f1000research.3-55.v1))

**First indexed:** 17 Mar 2014, 3:55 (doi: [10.12688/f1000research.3-55.v1](https://doi.org/10.12688/f1000research.3-55.v1))

## Commentary

In a recent press release (<http://www.nih.gov/news/health/dec2013/od-09.htm> December 9<sup>th</sup> 2013) on the occasion of the naming of Dr. Philip E. Bourne as the US NIH's first Associate Director for Data Science, NIH director Francis S. Collins, said that “*the era of ‘Big Data’ has arrived, and it is vital that the NIH play a major role in coordinating access to and analysis of many different data types that make up this revolution in biological information*”. We predict that one of Dr. Bourne's main priorities will be dissemination and visualisation of biological data through the web. Web pages are ideal tools for the dissemination of results and data. Dynamic interactivity is crucial in the discovery process, particularly for data-rich applications, as is the case of many websites that provide interfaces to biological databases. Databases storing genomic and other types of data have proliferated in the biological sciences, making them a data-rich, data-intensive set of disciplines. The visualisation of these data plays a crucial role in their interpretation as it permits the ability to hide or to focus on a particular detail, enabling the researcher to shed light on specific hypotheses or to create new ones based on observed patterns. The sheer complexity of biological data, however, requires more complex technologies than the usual static pages when accessing them. They require dynamic visualisation tools to allow real-time interactions and the usability of Web 2.0-based technologies.

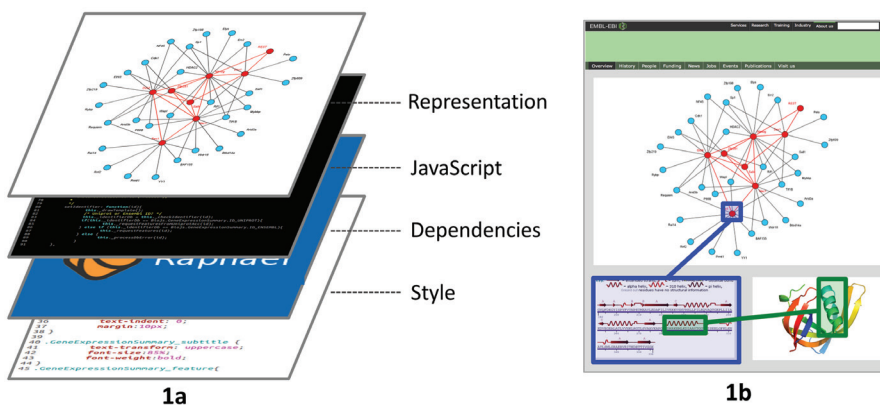
The JavaScript language as implemented in browsers is today's language for the web and has transformed modern applications into client-side browser-based. JavaScript offers common Application Programming Interfaces (API) purposely built for retrieval of remote data via RESTful services, making real-time interactivity possible. JavaScript has thus drastically changed the developer/application environment, gaining competitive advantage compared to other languages. To date, there have been successful initiatives for other languages to unify open source efforts, including BioPerl (Stajich *et al.*, 2002), BioJava (Prlic *et al.*, 2012) or BioRuby (Goto *et al.*, 2010). Non-language specific communities like the GMOD (Generic Model Organism Database; <http://www.gmod.org/>) have

also enjoyed wide adoption. These initiatives provide a centralised location in which to discover available functionality, facilitating the task of finding the desired functionality.

To cater to the burgeoning JavaScript community of developers for life sciences, we created BioJavaScript (Gómez *et al.*, 2013). BioJavaScript, or BioJS for short, is a framework designed for the development and sharing of biological component visualisation on the web using JavaScript. BioJS provides a catalogue or registry to enable the user or developer to find existing functionality for reuse. Components in the registry show their current maintenance status, the contact name of the main author and a guide showing how to install, customise and extend the component. BioJS, however, is not just a repository with a set of libraries; it is also a standard set of minimum guidelines for developers to reuse and create functionality in JavaScript as applied to biological concepts. BioJS thus provides a way for developers to build, extend and share functionality. Functionality in BioJS can be thought of as LEGO® pieces that can be connected to construct more complex applications.

## The BioJS project

BioJS allows developers and users alike to i) discover functionality through its registry, currently hosted at the European Bioinformatics Institute, ii) test in the registry itself the functionality of available components, iii) reuse components in different projects and applications, iv) combine components through a common set of predefined interactions and APIs, v) customise options for each component, vi) extend the functionality of available components in a standard manner - once the developer learns how to extend a component, extension of other components is consistent, vi) maintain components via community support and documentation and vii) develop new functionality following a predefined structured architecture common to all components. The typical component contains several layers of abstraction, including a style sheet, occasionally some dependencies to libraries such as D3.js, jQuery or Raphaël, the JavaScript layer and the representation layer (Figure 1a). Via common JavaScript event functions, several components can



**Figure 1.** **1a** shows the different layers that a BioJS component is divided into. The representation layer sits on top of the JavaScript layer, which similarly possesses a layer of dependencies and a style. **1b** presents an example of interactivity between three components, a protein-protein interaction network viewer, a secondary structure viewer and a tertiary structure viewer. Proteins in the network are represented as nodes and their interactions as edges. Clicking on a node makes the secondary and tertiary structure viewers retrieve the same protein. It is possible to select a secondary structure element in the 2D viewer and see where it is located in the 3D visualisation component.

be integrated in a single web page, allowing for interactivity among them. For example, a three-component frame (Figure 1b) can have a network visualisation component, with nodes representing proteins and edges representing interactions. Similarly, an alpha helix may be highlighted in a second component on the webpage that may cause a third component to highlight where the alpha helix is located in the tertiary structure of the same protein.

### The BioJS registry

At the time of writing (12-02-2014), the BioJS registry contains 39 components. The registry (<http://www.ebi.ac.uk/tools/biojs/>) constitutes one of the main BioJS access portals. It contains links to documentation, the community, tutorials and the list of components. A 'components' page in the registry contains a current list of components available. This list includes details of the component functionality, author and current version. By clicking on one of the component links, a new page appears with a series of tabs and interactive widgets to allow the user or developer to get a feel for how the component works and looks and to find details of its installation procedure, customisation and the methods that are part of the component's architecture, dependencies and events. Event functions can be tested on the page by using the text-box field provided. This is ideal to get a quick look and feel for the component, and it

is one of the most appreciated features of the BioJS project as it makes it possible for users to test a component without needing to install it.

Many of the components available in the registry have been developed for particular databases. The *ExpressionAtlasBaselineSummary* component is an example of this (Figure 2). This component was developed for the Expression Atlas database (<http://www.ebi.ac.uk/gxa/home>) and subsequently deposited in the BioJS registry for reutilisation. One of the great attractive features of BioJS component visualisation is that, by sharing the component through the registry, the Expression Atlas database will ensure that other users who utilise this component will be able to enjoy a visualisation of the data exactly as intended. Having a component that can be reutilised by remote websites allows the expression data contained in this database to always be shown in the same way. This consistency in the visualisation of the same biological concepts in the same manner by different resources facilitates discovery and ease of learning by end users.

### The BioJS community

There are different degrees of involvement that users may have in BioJS. Most will utilise the web components and the registry.

The screenshot shows the BioJS registry interface for the *ExpressionAtlasBaselineSummary* component. The main content area displays the following information:

**E-MTAB-513**  
 RNA-Seq of human individual tissues and mixture of 16 tissues (Illumina Body Map)  
 Organism(s): *Homo sapiens*  
 Reference(s): 22496456 22955988 23258890

Showing 1 of 1 genes found:

Gene	adipose	adrenal	brain	breast	colon	heart	kidney	leukocyte	liver	lung	lymph node	ovary	prostate	skeletal muscle	testis	thyroid
MT-ATP6		█						█		█	█					█

The sidebar on the right contains the following sections:

- Events**
  - onError
- Methods**
  - setQuery
  - setQuery (2)
  - setQuery (3)
  - setQuery (4)
- Dependencies**
  - JQuery Core 1.9.1

**Figure 2.** Screenshot of the *ExpressionAtlasBaselineSummary* component page in the BioJS registry, the Expression Atlas component for displaying baseline expression of genes based on RNA-seq experiments in the Expression Atlas database. Shown here is the transcription profiling by high throughput sequencing of RNA from individual and a mixture of 16 types of human tissues (E-MTAB-513 – Illumina Body Map). By default the 'Overview' tab is selected when accessing a registry component. On the right hand panel the prospective developer can test the main event functions (e.g., onError) and methods (e.g., setQuery). The installation tab provides the snippets of code required to install the component. Just copying and pasting this code should be sufficient for the component to work. The remaining tabs provide a more detailed explanation on the different options, methods and events specified for this component's usage.

For those who are involved in developing biological JavaScript applications, we expect that many may find the BioJS community an attractive place to meet like-minded developers and the right environment to share their work and seek feedback and/or support. There are several mail lists currently available that reflect different levels of involvement, such as those for developers and for the Steering Committee. We also have a Twitter account (@BiojsLibrary) that informs users of news and developments. We encourage anyone interested to become involved in the way that best suits them. A number of tutorials and workshops have been organised at the European Bioinformatics Institute and elsewhere, such as the VizBi conference. We plan to organise tutorials wherever they are of service and welcome BioJS developers to become tutors whenever they wish. We have a monthly call where the Steering Committee meets, and task forces are established to reflect the needs of developers and users. Examples of task forces that have been organised include those to discuss licensing issues, compatibilities between different dependencies, and funding and usability issues. We have strong ties with the Software Sustainability Institute (SSI), a UK-based organisation that aims to provide open source community support and sustainability. SSI has recently awarded the BioJS project several months of free consultancy to make it a more robust, well-maintained and useful resource.

## Discussion

The main motivation behind BioJS is to facilitate the creation, reutilisation and sharing of JavaScript functionality across the biological domain. BioJS provides a set of minimal common guidelines and a code architecture that makes creation of new functionality more efficient and consistent. The modular structure of components makes it possible for a web framework to isolate the visualisation aspect, and thereby facilitate maintainability. The documentation that each component is required to have is based on an automatic API generator, which transforms embedded code comments (required) into a structured document. The common architecture of components makes it possible to extend them in a consistent manner: once the developer learns how to create or extend a component, the generation or extension of new ones should be straightforward. BioJS allows developers to share the development of components by making use of the community's support. Components, once they are developed, can also be shared through the registry, so their exposure is likely to be increased. BioJS ensures that identical biological entities are visualised in the same manner and avoids different resources displaying the same content slightly differently. This reduces end-user confusion and ensures consistency across different biological domains. Having a common way in which biological entities are shown makes it easy for users to interpret the data in a more intuitive manner. BioJS can be particularly attractive to institutions that might want to have a common "flavour" for how they visualise a particular type of data. BioJS also encourages developers to aim at developing common implementations by following common guidelines on how to implement code, and makes people aware of what components are being developed by an organisation. This may constitute a competitive advantage for showcasing

development contributions carried out by a particular research group or institution.

## Future directions

BioJS is a thriving community that so far has been able to attract plenty of voluntary contributions in the same spirit as other biologically-inspired open-source communities. We aim to start a series of Google Summer of Code projects to attract students interested in developing their JavaScript skills while working on life-sciences research projects. A collaborative research project has been established with the bioinformatics consultancy Eagle Genomics to develop functionality that support visualisation of genome data without a reference assembly. Many important projects and institutions have already shown commitment to the project by developing components (i.e. SwissProt ([UniProt Consortium, 2014](#)), Genome3D ([Lewis et al., 2013](#)), Reactome ([Croft et al., 2011](#)), Expression Atlas ([Petryszak et al., 2014](#)), TGAC Browser (<http://tgac-browser.tgac.ac.uk/>), etc.), and the time looks ripe to take the project to a new phase. Many challenges remain, however. The BioJS project is planning to establish a sustainable future with both robust institutional and community financial support. As the number of users increases, the need for support increases, both in terms of support to help contributors deliver but also to keep track of the state of maintenance of deposited components. Currently most of the work is done by volunteers who help fix bugs and improve the quality of submitted work. BioJS has worked well as a prototype for many projects where simple components have been created. Our first-stage mission of developing a common framework has thus been achieved. Whether we are able to meet the expectations raised by the potential of the project will only be guaranteed by the explicit commitment of important players in the bioinformatics arena.

---

## Author contributions

All authors have participated in the development of the BioJS community through provision of code, meeting attendance or writing of grants.

## Competing interests

No competing interests have been disclosed.

## Grant information

We are grateful to the NHLBI Proteomics Center Award (HHSN268201000035C), the UK's Biotechnology and Biological Sciences Research Council (BBSRC) and European Commission grant PSIMEx (FP7-HEALTH-2007-223411).

*The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

## Acknowledgements

We would like to thank Teresa K. Attwood for comments on the manuscript.

## References

---

Croft D, O'Kelly G, Wu G, *et al.*: **Reactome: a database of reactions, pathways and biological processes.** *Nucleic Acids Res.* 2011; **39**(Database issue): D691–D697.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Goto N, Prins P, Nakao M, *et al.*: **BioRuby: bioinformatics software for the Ruby programming language.** *Bioinformatics.* 2010; **26**(20): 2617–2619.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Lewis TE, Sillitoe I, Andreeva A, *et al.*: **Genome3D: a UK collaborative project to annotate genomic sequences with predicted 3D structures based on SCOP and CATH domains.** *Nucleic Acids Res.* 2013; **41**(Database issue): D499–D507.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Petryszak R, Burdett T, Fiorelli B, *et al.*: **Expression Atlas update—a database**

**of gene and transcript expression from microarray- and sequencing-based functional genomics experiments.** *Nucleic Acids Res.* 2014; **42**(1): D926–D932.

[PubMed Abstract](#) | [Publisher Full Text](#)

Prlc A, Yates A, Bliven SE, *et al.*: **BioJava: an open-source framework for bioinformatics in 2012.** *Bioinformatics.* 2012; **28**(20): 2693–2695.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Stajich JE, Block D, Boulez K, *et al.*: **The Bioperl toolkit: Perl modules for the life sciences.** *Genome Res.* 2002; **12**(10): 1611–1618.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

UniProt Consortium: **Activities at the Universal Protein Resource (UniProt).**

*Nucleic Acids Res.* 2014; **42**(1): D191–D198.

[PubMed Abstract](#) | [Publisher Full Text](#)

## Current Referee Status:

---

### Referee Responses for Version 1



**Reinhard Schneider**

University of Luxemburg, Luxemburg, USA

**Approved: 17 March 2014**

**Referee Report:** 17 March 2014

**doi:**[10.5256/f1000research.3850.r3706](https://doi.org/10.5256/f1000research.3850.r3706)

Visualization is one of the bottlenecks we are facing in modern biology. With bigger and bigger data sets originating from various heterogeneous sources that span several concepts, dimensions and timelines, it is becoming an art to condense the wealth of information into good and easy ways to capture visual pictures.

For non-expert users it can be quite a challenge to come up with a "publication ready" visualization of complex relationships. The BioJS community tries to tackle this challenge by employing an open modular architecture, which allows a broad audience to use and further develop the components in the registry. The community already provides a wide spectrum of available components and it will be very interesting to see the future development of this open source community.

**I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.**

**Competing Interests:** No competing interests were disclosed.



**Phil Bourne**

National Institutes of Health (NIH), Bethesda, ML, USA

**Approved: 20 February 2014**

**Referee Report:** 20 February 2014

**doi:**[10.5256/f1000research.3850.r3819](https://doi.org/10.5256/f1000research.3850.r3819)

I am supportive of the BioJS registry with regards to the technology used and its principles. To give you a flavour of what I will be looking for, from an NIH perspective, is really how much the components are used and why. What findings have resulted in their use? If there is non-use of components and why etc. In short, I will be looking more at the degree of software projects that are making a difference to the scientific community. That said, I very much look forward to seeing how the BioJS community develops.

**I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.**

**Competing Interests:** No competing interests were disclosed.

---