



WEB TOOL

Sequence, a BioJS component for visualising sequences [v1; ref status: indexed, <http://f1000r.es/2v9>]

John Gomez, Rafael Jimenez

European Bioinformatics Institute EMBL-EBI, Hinxton, CB10 1SD, UK

v1 First published: 13 Feb 2014, 3:52 (doi: [10.12688/f1000research.3-52.v1](https://doi.org/10.12688/f1000research.3-52.v1))
 Latest published: 13 Feb 2014, 3:52 (doi: [10.12688/f1000research.3-52.v1](https://doi.org/10.12688/f1000research.3-52.v1))

Abstract

Summary: Sequences are probably the most common piece of information in sites providing biological data resources, particularly those related to genes and proteins. Multiple visual representations of the same sequence can be found across those sites. This can lead to an inconsistency compromising both the user experience and usability while working with graphical representations of a sequence. Furthermore, the code of the visualisation module is commonly embedded and merged with the rest of the application, making it difficult to reuse it in other applications. In this paper, we present a BioJS component for visualising sequences with a set of options supporting a flexible configuration of the visual representation, such as formats, colours, annotations, and columns, among others. This component aims to facilitate a common representation across different sites, making it easier for end users to move from one site to another.

Availability: <http://www.ebi.ac.uk/Tools/biojs>;
<http://dx.doi.org/10.5281/zenodo.8299>

This article is included in the [BioJS Collection](#)**Article Status Summary****Referee Responses**

Referees	1	2
v1 published 13 Feb 2014	 report	 report

1 **Christoph Gille**, University Medicine
Berlin (Charité) Germany

2 **Jeremy Goecks**, George Washington
University USA

Latest Comments

No Comments Yet

Corresponding author: John Gomez (jgomez@ebi.ac.uk)

How to cite this article: Gomez J and Jimenez R. *Sequence, a BioJS component for visualising sequences [v1; ref status: indexed, <http://f1000r.es/2v9>]* F1000Research 2014, 3:52 (doi: [10.12688/f1000research.3-52.v1](https://doi.org/10.12688/f1000research.3-52.v1))

Copyright: © 2014 Gomez J and Jimenez R. 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: NHLBI Proteomics Center Award HHSN268201000035C.

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 were disclosed.

First published: 13 Feb 2014, 3:52 (doi: [10.12688/f1000research.3-52.v1](https://doi.org/10.12688/f1000research.3-52.v1))

First indexed: 17 Mar 2014, 3:52 (doi: [10.12688/f1000research.3-52.v1](https://doi.org/10.12688/f1000research.3-52.v1))

Introduction

Visualising biological data on the web is a common practice on sites providing bio-oriented services and resources. A wide variety of JavaScript libraries are being used to build pieces of software capable of representing bio-entities such as DNA sequences¹, protein sequences (<http://www.uniprot.org>), protein structures (<http://www.rwpdb.org>), ontology trees², protein-protein interactions (<http://www.ebi.ac.uk/intact/>)³, and others. Therefore, a variety of possible visual representations for the same bio-entity can be found as a result of its multiple implementations. In many cases, such implementations are difficult to maintain, test, and reuse as they are developed only with one use case in mind. Furthermore, user experience (UX) and usability across different sites may be compromised.

One particular type of data commonly affected by multiple representations is the sequence, either a DNA or protein sequence. A sequence is a common bio-entity present in most sites offering biological data resources. **Figure 1** shows different visual representations of a protein sequence as it can be found in Uniprot (<http://www.uniprot.org>), Dasty⁴ (<http://www.ebi.ac.uk/dasty>) and Ensembl (<http://www.ensembl.org>), among others^{5,6}. Multiple features are identified across the entire set of sequences. Features such as formatting, indexing numbers, annotations, marks, colouring tags, and even the capability of user interaction are not integrated in one reusable piece of code. Instead, multiple representations prevail. Furthermore, web developers often make their own isolated efforts to reproduce those views for their sites and, in most cases, the representation is not identical, no documentation is available, and often they are not portable to other sites.

In this paper, a reusable component to visualise sequences is presented under the BioJS set of minimum standards for visualisation of biological components. BioJS is a community-driven standard to develop visualisation functionality⁷. The library is developed using well-established methodologies and object-oriented design

with inheritance that facilitates rapid development, reuse, extension, integration and deployment of web applications.

The Sequence component

Exploring sequence visualisation across different sites reveals a set of features that should be supported by a single, reusable, and well documented piece of code, capable of painting sequences on the web in a consistent manner. In this sense, BioJS provides a baseline for Javascript coding and development to create pieces of reusable code, called components. Creating a new *Sequence* component consists of extending a core BioJS class and defining three core concepts: options, methods and events. Options are the data required by the component for initialisation, while methods and events are actions supported in execution time. Methods are fired externally while events are triggered in the component and exposed to external listeners.

Methods and events allow the component to communicate with others components as well as web applications. **Figure 2** shows a working example implemented within the Biotea project⁸. This example shows a communication between two component instances, the *Sequence* component and the *Protein3D* component. When a region (highlighted in yellow) on the sequence is selected, automatically a selection action is fired in the Protein3D. Additionally, *Sequence* supports a set of options to change the visual representation of the sequence by using different formats, colours, indexing numbers, annotations and more. It helps deployment because the component can be easily fitted to the particular need. **Figure 3** shows an example of the *Sequence* component displaying the protein P918283 in CODATA format.

As any other BioJS component, the *Sequence* component is well documented and has been tested during development, not only for functionality but also for usability. BioJS makes it easier to document the code by adding annotations that are later exposed as a web page. Thus, human-friendly documentation is generated without



Figure 1. Multiple representations compiled as one flexible BioJS component.

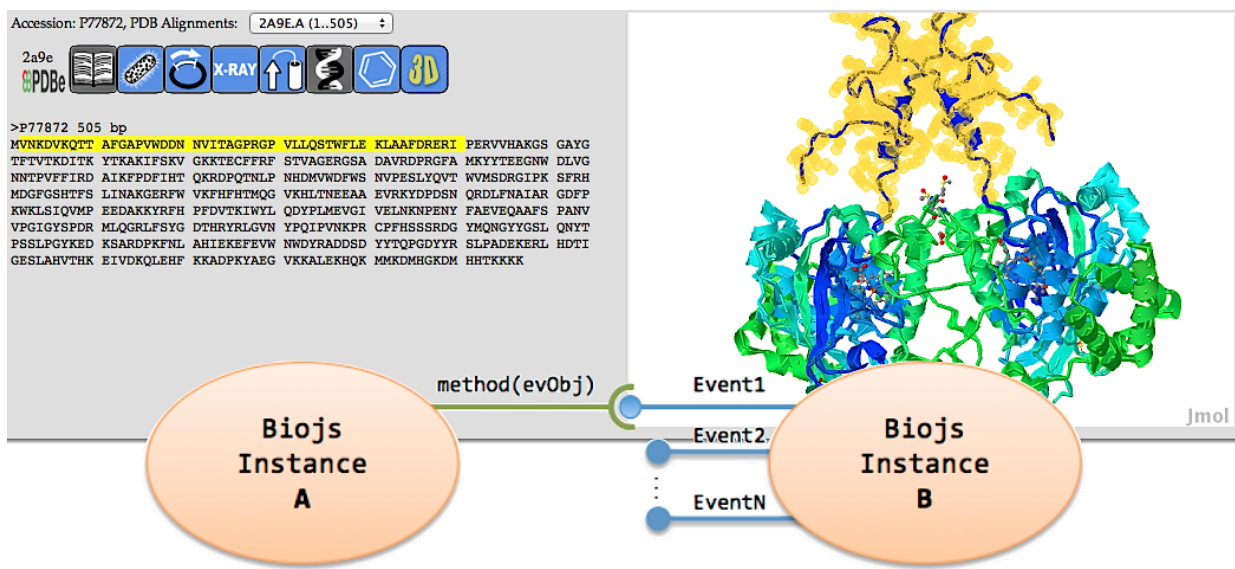


Figure 2. Example of communication between Sequence and Protein3D components.

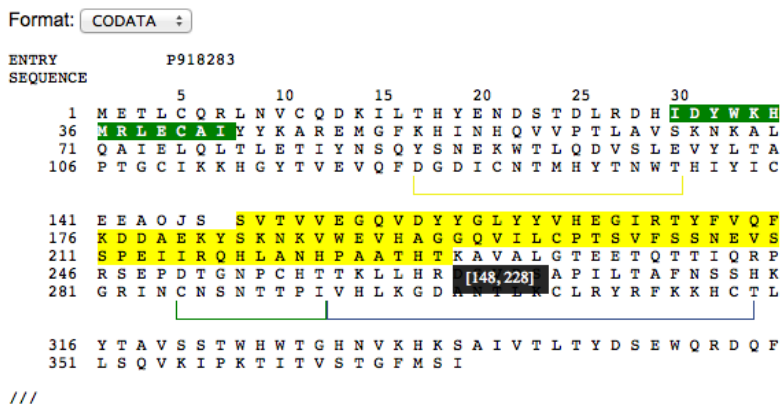


Figure 3. Example displaying the sequence corresponding to the UniProt accession P918283. The part highlighted in yellow denotes the current selection, the black pop-up box indicates what the interval is with every move of the pointer. Green highlight denotes an annotation on that interval. Multiple annotations are supported.

any additional effort. BioJS web pages for components are compiled in a registry that acts as a showcase of working examples extracted from the component annotations. The registry makes it easier for both developers and end users to understand components and their functionality. Once a component has met the BioJS guidelines, it becomes a candidate to be submitted and publicly shared in the common repository of components, the EBI BioJS registry (<http://www.ebi.ac.uk/Tools/biojs/registry/>). There, it is possible to find more information about options, installation, methods, and events (<http://www.ebi.ac.uk/Tools/biojs/registry/Biojs.Sequence.html>).

Future work

Currently, the *Sequence* component supports the visualisation of a single strand. However, in some cases, it should be more interesting to display similarities between two or multiple sequences. Another possible extension is using this component as a base for multiple aligned sequences visualisation. Aligner algorithms⁹ could be run

on the server side or consumed from a web service¹⁰ while the component would be in charge of painting the similarities, taking advantage of already developed features such as colouring, highlighting, and tagging.

Collaborative work and social networking is nowadays a mechanism for knowledge construction. Such features can be integrated into the *Sequence* component so end users can submit sequences and annotations to public sequence databases such as UniProt. Comments and references could also be added, adding valuable information for a researcher during his/her investigation.

Software availability

Zenodo: Sequence BioJS component for visualising sequences, doi: [10.5281/zenodo.829911](https://doi.org/10.5281/zenodo.829911).

GitHub: BioJS, <http://www.ebi.ac.uk/Tools/biojs>.

Author contributions

The work presented here was carried out in collaboration between both authors. RJ collected the component requirements across several EBI teams and collaborated with JG in the visual design, UX and usability tests. JG implemented all functionality in JavaScript following the guidelines of BioJS. This manuscript was written and revised by both authors.

Competing interests

No competing interests were disclosed.

Grant information

NHLBI Proteomics Center Award HHSN268201000035C.

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

Acknowledgements

The authors thank Henning Hermjakob for his support to the project, and Leyla Garcia for his comments on the component. We also acknowledge Sangya Pundir for helpful UX and usability testing and invaluable feedback.

The authors thank all researchers who have deposited information into publically available datasets as well as developers who have provided their work as open source: our work stands upon their shoulders and would not have been possible without them.

References

- Rutherford K, Parkhill J, Crook J, *et al.*: **Artemis: sequence visualization and annotation.** *Bioinformatics.* 2000; **16**(10): 944–945.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Cote RG, Jones P, Apweiler R, *et al.*: **The Ontology Lookup Service, a lightweight cross-platform tool for controlled vocabulary queries.** *BMC Bioinformatics.* 2006; **7**(1): 97.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Kerrien S, Aranda B, Breuza L, *et al.*: **The IntAct molecular interaction database in 2012.** *Nucleic Acids Res.* 2012; **40**(Database issue): D841–D846.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Villaveces JM, Jimenez RC, Garcia LJ, *et al.*: **Dasty3, a WEB framework for DAS.** *Bioinformatics.* 2011; **27**(18): 2616–2617.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Ilyin VA, Pieper U, Stuart AC, *et al.*: **ModView, visualization of multiple protein sequences and structures.** *Bioinformatics.* 2003; **19**(1): 165–166.
[PubMed Abstract](#) | [Publisher Full Text](#)
- O'Shea JP, Chou MF, Quader SA, *et al.*: **pLogo: a probabilistic approach to visualizing sequence motifs.** *Nat Methods.* 2013; **10**(12): 1211–1212.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Gómez J, García LJ, Salazar GA, *et al.*: **Biojs: an open source JavaScript framework for biological data visualization.** *Bioinformatics.* 2013; **29**(8): 1103–1104.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- García A, García LJ, Gómez J: **Conceptual exploration of documents and digital libraries in the biomedical domain.** In volume 952 of *Semantic Web Applications and Tools for Life Sciences*,. Adrian Paschke *et al.*, editor. 2012, Springer: France.
[Reference Source](#)
- Li H, Homer N: **A survey of sequence alignment algorithms for next-generation sequencing.** *Brief Bioinform.* 2010; **11**(5): 473–483.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Larkin MA, Blackshields G, Brown NP, *et al.*: **Clustal w and clustal x version 2.0.** *Bioinformatics.* 2007; **23**(21): 2947–2948.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Gomez J, Jimenez R: **Sequence BioJS component for visualising sequences.** *Zenodo.* 2014.
[Data Source](#)

Current Referee Status:

Referee Responses for Version 1



Jeremy Goecks

Computational Biology Institute, George Washington University, Washington, DC, USA

Approved: 17 March 2014

Referee Report: 17 March 2014

doi:[10.5256/f1000research.3717.r3802](https://doi.org/10.5256/f1000research.3717.r3802)

Here, the authors present *Sequence*, a web-based visualization component for biological sequence data implemented in JavaScript. Investigators can use *Sequence* to visualize both DNA and protein sequences, either as a standalone visualization or together with other visualizations.

Strengths of *Sequence* include (a) the ability to customize sequence using options and (b) integration of sequence via events. These features ensure that *Sequence* can be used in a wide variety of applications.

What is missing from this manuscript is a description of how well *Sequence* scales to large sequences and whether a *Sequence* visualization can be updated dynamically in response to events from other components.

Overall, *Sequence* is a solid contribution to web-based visualization that is useful as it is and forms the foundation for more complex web-based sequence visualization in the future.

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.



Christoph Gille

Computational Biochemistry Group, Institute of Biochemistry, University Medicine Berlin (Charité), Berlin, Germany

Approved: 17 March 2014

Referee Report: 17 March 2014

doi:[10.5256/f1000research.3717.r3694](https://doi.org/10.5256/f1000research.3717.r3694)

General

The authors present the first re-usable JavaScript based sequence component. It can be used in web applications dealing with bio-polymers like proteins and nucleotide sequences and can also interact with other parts of the website via events.

Previously, Java applets have been used for interactive web content. However, Java constitutes an additional layer of software and thereby carries an own set of technical problems and risks. For this reason,

JavaScript is being increasingly used at the client side. In this respect, the development of BioJS components follows a general trend.

The BioJS registry is the first and only framework plus standard for interactive web components, and *Sequence* will be one of the most important components following the BioJS specification. Therefore, I expect that the *Sequence* component will be widely used in bioinformatics web services. Even if current features might not satisfy all needs, the BioJS format allows for extensions and incorporation of new features with the source code clear and well documented, allowing developers to change it to their requirements.

Manuscript

I would suggest replacing the word "compiled" with another word (in the figure 1 legend and in the third paragraph of the *Sequence* component section) as it might be mistaken for source code getting compiled on a server like on the Debian Linux server.

The manuscript does not provide answer to some important questions:

1. Is the length of the sequence limited?
2. Is the sequence immutable? Or could it change like alternative splicing? Can parts of the sequences be hidden like cutting of signal peptide?
3. "Indexing numbers" - does the numbering support PDB insertion codes?

It would be good if these points could be clarified in the manuscript.

Example

For demonstration, the authors have coupled the sequence view with a BioJS 3D component. With the newest Java, the JMol applet fails to start with the message: "*Your security system has blocked an untrusted ...*". I expect that the line Permissions: sandbox in the jar-file manifest and signing the jar-file will fix the problem.

The authors should also consider using a JavaScript based 3D visualization.

API

On events like 'Annotation Clicked', there is no parameter indicating whether the context pop-up trigger (right click, long touch) is active and what modifier keys like Shift and Ctrl are pressed - this should be made clearer for ease of use.

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.
