WEB TOOL

treeWidget: a BioJS component to visualise phylogenetic trees

[version 1; peer review: 1 approved, 1 approved with reservations]

Fabian Schreiber^{1,2}¹The Wellcome Trust Sanger Institute, Hinxton, Cambridge, CB10 1SD, UK²European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK

V1 First published: 13 Feb 2014, 3:49
<https://doi.org/10.12688/f1000research.3-49.v1>
Latest published: 13 Feb 2014, 3:49
<https://doi.org/10.12688/f1000research.3-49.v1>

Abstract

Summary: Phylogenetic trees are widely used to represent the evolution of gene families. As the history of gene families can be complex (including lots of gene duplications), its visualisation can become a difficult task. A good/accurate visualisation of phylogenetic trees - especially on the web - allows easier understanding and interpretation of trees to help to reveal the mechanisms that shape the evolution of a specific set of gene/species. Here, I present *treeWidget*, a modular BioJS component to visualise phylogenetic trees on the web. Through its modularity, *treeWidget* can be easily customized to allow the display of sequence information, e.g. protein domains and alignment conservation patterns.

Availability: <http://github.com/biojs/biojs>;
<http://dx.doi.org/10.5281/zenodo.7707>

Open Peer Review

Approval Status ? ✓

	1	2
version 1 13 Feb 2014	? view	✓ view

1. **Jaime Huerta-Cepas**, EMBL European Bioinformatics Institute, Heidelberg, Germany

2. **Stephen A. Smith**, University of Michigan, Ann Arbor, MI, USA

Any reports and responses or comments on the article can be found at the end of the article.

BIOJS

This article is included in the **BioJS** collection.

This article is included in the **Phylogenetics** collection.

EMBL-EBI

This article is included in the **EMBL-EBI** collection.

Corresponding author: Fabian Schreiber (fs@ebi.ac.uk)

Competing interests: No competing interests were disclosed.

Grant information: Wellcome Trust [WT077044/Z/05/Z] to FS.

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

Copyright: © 2014 Schreiber F. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), 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).

How to cite this article: Schreiber F. *treeWidget: a BioJS component to visualise phylogenetic trees* [version 1; peer review: 1 approved, 1 approved with reservations] F1000Research 2014, 3:49 <https://doi.org/10.12688/f1000research.3-49.v1>

First published: 13 Feb 2014, 3:49 <https://doi.org/10.12688/f1000research.3-49.v1>

Introduction

A phylogenetic tree is a branching diagram showing the inferred relationships of genes or species. Reconstructing a phylogenetic tree is a routine task in most evolutionary-related analyses and a number of databases exist containing precomputed phylogenetic trees (e.g. TreeFam¹, Ensembl Gene Trees², Panther³). These reconstructed trees can vary considerably in the number of gene/species shown and in their complexity. While there are many offline tools available to visualise phylogenetic trees (e.g. ETE⁴), the number of online tools for this purpose is rather limited. Some of them are written in Java and tend to become slow when the number of nodes/edges increases (e.g. Archaeopteryx⁵). Others written in JavaScript, are therefore faster and more scalable, but do not allow the additional display of useful sequence annotation, e.g. protein domains and alignments. (e.g. phyloWidget⁶, jsPhyloSVG⁷). Yet other tools offer that functionality but are not available for download, customization or to be embedded into the users' own websites (e.g. iTol⁸). Despite its widespread use in Bioinformatics, biological web applications for viewing phylogenetic trees are usually implemented with no standard reutilisation guidelines in mind.

Here I present *treeWidget*, a BioJS⁹ component written in JavaScript to visualise phylogenetic trees on the web. *treeWidget* can be easily integrated into websites and customized. To my knowledge, this is the first modular tree visualisation component.

The *treeWidget* component

The *treeWidget* component has been developed as part of the TreeFam project and follows the standards set by the BioJS registry (¹, see Figure 2 for an example).

The BioJS registry is a centralised repository of BioJS components hosted at the European Bioinformatics Institute. *treeWidget* uses the JavaScript library D3 for building trees¹⁰. It reads trees in JSON format (<http://www.json.org/>) and plots them as scalable vector graphics (SVG, see the *treeWidget* component documentation for a working example of a JSON-formatted tree). Additionally, *treeWidget* can plot annotation in a separate SVG. This way the tree diagram stays fixed whenever the annotation changes (e.g. switching from

a domain to a sequence alignment view). *treeWidget* automatically scales the SVG according to the tree's height and so will work with small trees containing only 3 leaves up to trees with several hundreds of leaves by plotting the tree size according to the tree height (number of internal nodes).

To make the leaf names of gene trees more meaningful, users can add additional information (e.g. gene name, source species, common name, gene function, etc) to the JSON file. *treeWidget* will then plot this information next to the tree leaves (see Figure 2). *treeWidget* can either draw an ultrametric tree (all leaves have the same branch length) or a tree with estimated branch lengths and inner nodes can be labelled with taxon names and bootstrap values.

In cases where the gene trees contain too many leaves, it is useful to start looking at the gene tree by focussing on a specific part of the tree. This could be, for example, a pig gene that the user is particularly interested in. *treeWidget* allows highlighting of a specific gene and collapsing of sister clades to hide less relevant parts of the tree display (see Figure 1). These collapsed clades can be expanded/collapsed by a mouse click. To make the identification of related species in a gene tree easier, *treeWidget* colours pre-selected taxonomic clades (see Figure 2).

Figure 1. Gene tree with focus on the CCNC gene in pig. The surrounding sister clades (Carnivora, Cetartiodactyla, Laurasiatheria) are collapsed. They can be expanded by clicking on the node.

Figure 2. The *treeWidget* view showing a part of the *BRCA2* gene tree with clades coloured according to taxonomy.

Application

treeWidget can be used to visualise the evolution of species but also that of genes as it is done on the TreeFam website (<http://www.treefam.org>). TreeFam's main goal is to present phylogenetic trees of gene families across the animal tree of life. TreeFam also predicts orthology/paralogy relationships: we speak of orthologs when two genes in different species are the result of a speciation event, whereas paralogs are genes stemming from a duplication event. The *treeWidget* component allows the display of this information by labelling the internal nodes of each gene tree as either speciation or duplication events. Additionally, the *treeWidget* component displays patterns of alignment conservation as well as matches of Pfam¹¹ protein domains on each sequence in the database. In Figure 2 the conserved alignment pattern from the underlying protein sequence alignment is shown. The white alignment parts represent gaps and green parts are aligned parts. Visualising alignment conservation in conjunction with Pfam domains along a gene tree gives useful insights about the evolution of domain architectures¹². Furthermore, this view can be used to spot problems with assembled genes (split genes or falsely assembled genes).

Conclusions

The *treeWidget* component provides a platform for the exploration of complex phylogenetic trees depicting the evolution of large gene

families with many duplication/speciation events as well as displaying sequence annotation features. Visualising such complex data allows researchers to see interesting features for further study. I expect this component to be particularly useful to developers and users alike, requiring little technical knowledge for its full functioning.

Software availability

Zenodo: BioJS TreeWidget component, doi: [10.5281/zenodo.7751](https://doi.org/10.5281/zenodo.7751)¹³

GitHub: BioJS, <http://github.com/biojs/biojs>

Competing interests

No competing interests were disclosed.

Grant information

Wellcome Trust [WT077044/Z/05/Z] to FS.

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

Acknowledgements

The author thanks Mateus Patricio, Miguel Pignatelli, Matthieu Muffato as well as Alex Bateman for useful feedback.

References

- Schreiber F, Patricio M, Muffato M, *et al.*: **TreeFam v9: a new website, more species and orthology-on-the-fly**. *Nucleic Acids Res.* 2014; **42**(Database issue): D922–D925.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Vilella AJ, Severin J, Ureta-Vidal A, *et al.*: **EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates**. *Genome Res.* 2009; **19**(2): 327–335.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Mi H, Muruganujan A, Thomas PD: **PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees**. *Nucleic Acids Res.* 2013; **41**(Database issue): D377–D386.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Huerta-Cepas J, Dopazo J, Gabaldón T: **ETE: a python Environment for Tree Exploration**. *BMC Bioinformatics.* 2010; **11**(1): 24.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Han MV, Zmasek CM: **phyloXML: XML for evolutionary biology and comparative genomics**. *BMC Bioinformatics.* 2009; **10**: 356.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Jordan GE, Piel WH: **PhyloWidget: web-based visualizations for the tree of life**. *Bioinformatics.* 2008; **24**(14): 1641–1642.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Smits SA, Ouverney CC: **jsPhyloSVG: a javascript library for visualizing interactive and vector-based phylogenetic trees on the web**. *PLoS One.* 2010; **5**(8): e12267.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Letunic I, Bork P: **Interactive Tree Of Life v2: online annotation and display of phylogenetic trees made easy**. *Nucleic Acids Res.* 2011; **39**(Web Server issue): W475–8.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free 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](#)
- D3.js - Data-Driven Documents**.
[Reference Source](#)
- Punta M, Coghill PC, Eberhardt RY, *et al.*: **The Pfam protein families database**. *Nucleic Acids Res.* 2012; **40**(Database issue): D290–D301.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Forslund K, Sonnhammer EL: **Evolution of protein domain architectures**. *Methods Mol Biol.* 2012; **856**(Chapter 8): 187–216.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Schreiber F: **BioJS TreeWidget component**. *Zenodo.* 2014.
[Data Source](#)

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 24 March 2014

<https://doi.org/10.5256/f1000research.3676.r4248>

© 2014 Smith S. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Stephen A. Smith

Ecology and Evolutionary Biology, University of Michigan, Ann Arbor, MI, USA

treeWidget is a nice feature full JavaScript tree viewer. This is a relatively straightforward announcement of the tool and I see no major problems. Below are some comments for the author's consideration.

Major

The discussion of the license is not in the manuscript. I believe it is Apache 2.0 but it would be good to mention that. Also, I understand it is open source but I might also mention that clearly as well.

I feel like the author overstates the originality of the library. There is mention of [jsphyloSVG](#) which of course is a modular tree visualisation component (negating the last sentence of the introduction). Furthermore, there are others. In particular, Roderic Page has been developing a number of different options that are simple and perhaps not as feature-full but serve to have trees in the BioNames project (<http://bionames.org/>). In fact, this has been around for a few years and available for experimentation (here is one of the early posts about this <http://iphylo.blogspot.com/2012/12/viewing-phylogenies-on-web-javascript.html>). Of course not all of these would be considered more than experimentation, but I would note that it has been an area of long discussion in the bioinformatics field for a while now.

The documentation is pretty difficult to navigate. There is a fair bit of (also unclear at this point) documentation of the BioJS library with some specific language that is not well defined (what is the registry for exactly?). If *treeWidget* is in fact going to be particularly useful for web developers to drop into a website, I would suggest making a tutorial specifically designed to show a developer how to incorporate just *treeWidget* and not the entire BioJS package.

I was unable to generate dynamic examples. This is related to the above comment, but I would generate some more examples and tutorials that explore these options in more detail. Please distribute the code used to make the figures as part of the distribution or in another package. That would be helpful.

Minor

I feel like a few edits would improve the manuscript quite a bit. For example, in the introduction, you state:

"Reconstructing a phylogenetic tree is a routine task in most evolutionary-related analyses and a number of databases exist containing precomputed phylogenetic trees (e.g. TreeFam¹, Ensembl Gene Trees², Panther³). These reconstructed trees can vary considerably in the number of gene/species shown and in their complexity"

For the first sentence, I am not entirely sure I understand the relevance of the existence of precomputed trees and if that is important, this is an odd assortment of those resources. Of course there are many more. For the second sentence, I am not certain what you mean by varying in their complexity. I don't think you mean in their complexity in computational sense (but perhaps you do?).

Because JSON is not often used in phylogenetics packages, I suspect that you will attract more users if you can take newick formats. You can see some examples using other packages here: <http://bl.ocks.org/rdmpage/raw/4224658/>.

Competing Interests: No competing interests were disclosed.

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

Reviewer Report 25 February 2014

<https://doi.org/10.5256/f1000research.3676.r3713>

© 2014 Huerta-Cepas J. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Jaime Huerta-Cepas

Structural and Computational Biology Unit, EMBL European Bioinformatics Institute, Heidelberg, Germany

TreeWidget is a handy JavaScript component for the visualization of phylogenetic data. It is currently used to display trees in the [TreeFam](#) database and allows for inline representation of rectangular trees, alignments and sequence domains. After testing the application with trees of different sizes I came out with the following comments that may serve the author to improve this publication:

Major***Usability***

Apart from domain and alignment annotation, the documented functionality of *TreeWidget* looks quite limited as compared with other JavaScript components. In my tests, I could only represent static trees with tip labels and motif structure. I have the impression that the application is capable of much more, but I could not find any reference to it in the help pages. For instance: Would it be possible to add or modify OnClick and MouseOver events? Search for nodes in large trees? What about zooming or showing bootstrap and branch length values? These are basic features present in the TreeFam version and in most tree viewers, but not in the current version of *TreeWidget*. Note that although Figure 1 shows bootstrap support values in the tree, I could not find that option in the documentation. It would be useful to have the example code that generates the figures in the paper.

JSON is not a common format for phylogenetic data, but it seems to be the only way to load trees into the application. Can *TreeWidget* read the newick or extended-newick formats? If so, please document it. In my case, I had to write a little parser to convert regular trees into treewidget/d3 format (gist.github.com/jhcepas/9205262) to test the software, but it would be great to have this feature as a built-in option. The same applies for alignments. I think that providing Newick and Fasta files would be much more convenient for end users than pointing them to create a new JSON structure on top of the original data.

Manuscript

Although I acknowledge the potential usefulness of *TreeWidget*, I think the text goes a bit too far when states that this is the "*first modular tree visualization component*". Even if we restrict to the JavaScript context, several libraries do provide online tree visualization from some time ago. In fact, the same d3 back-end library used by *TreeWidget* is a JavaScript modular library that supports tree and network visualization by itself (bl.ocks.org/mbostock/4063570). Of note, the nice performance and responsiveness obtained when drawing large trees is a d3 feature available for the representation of any type of hierarchical data. Similarly, *jsPhyloSVG* is a modular JavaScript library that allows online representation of phylogenetic trees, providing interesting features such as newick support and circular tree drawing. This should not prevent *TreeWidget* from being published in *F1000Research*, but the emphasis should be put on the possibility of representing phylogenetic information (i.e. duplication and speciation events) together with alignments and domain structure, in a very easy way.

There are also several mistakes regarding the literature cited in the introduction: 1) The ETE package is not only an offline tool, but a programming library offering a *webplugin* module that can be used for online interactive visualization of custom phylogenetic data (i.e. the *PhylomeDB* database uses ETE to render *interactive tree images with alignments and Pfam domain annotations*). 2) To my knowledge, *PhyloWidget* is not JavaScript but Java. 3) I would not say that *Archeopteryx* is that slow for medium/large trees, and it also offers web integration and alignment and domain visualization.

Minor

Documentation

While testing the library I had the feeling that the documentation was not very clear. First, I could not find the relevant code, documentation or examples within the github repository provided as the main link in manuscript. It seems that the so called '*registry installation*' step is necessary to start using the library. This will install the whole BioJS package, making the code and examples

available at `./biojs/target/registry/`, which is in fact the base path assumed by the examples provided in the online example. This is not a big deal, but I had to guess it by browsing different pages and files of the BioJS project. It would be useful to point readers to the specific installation and help pages of *TreeWidget*, clarifying whether it can be used independently or requires the BioJS project to be downloaded and compiled as a whole. Note, that it is also not clearly mentioned in the manuscript, that the examples and documentation of this component can be found at <https://www.ebi.ac.uk/Tools/biojs/registry/Biojs.Tree.html>.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research