WEB TOOL

**wigExplorer**, a BioJS component to visualise wig data [version 1; peer review: 1 approved with reservations, 1 not approved]

Anil S. Thanki\(^1\), Rafael C. Jimenez\(^2\), Gemy G. Kaithakottil\(^1\), Manuel Corpas\(^1\), Robert P. Davey\(^1\)

\(^1\)The Genome Analysis Centre, Norwich Research Park, Norwich, NR4 7UH, UK
\(^2\)European Bioinformatics Institute, Hinxton, Cambridge, CB10 1SD, UK

**Abstract**

**Summary:** *wigExplorer* is a BioJS component whose main purpose is to provide a platform for visualisation of wig-formatted data. Wig files are extensively used by genome browsers such as the UCSC Genome Browser. *wigExplorer* follows the BioJS standard specification, requiring a simple configuration and installation. *wigExplorer* provides an easy way to navigate the visible region of the canvas and allows interaction with other components via predefined events.

**Availability:** [http://github.com/biojs/biojs](http://github.com/biojs/biojs); [http://dx.doi.org/10.5281/zenodo.7721](http://dx.doi.org/10.5281/zenodo.7721)

This article is included in the Bioinformatics gateway.

This article is included in the EMBL-EBI collection.

This article is included in the BioJS collection.
Introduction
Numerous web applications exist for visualisation of biological data. Data can be prepared for visualisation using a variety of formats, one of which is the widely used wiggle (wig) file. A wiggle file contains text that defines either a feature or a data track. The wiggle format was developed by the UCSC genome browser\(^1\) and then quickly adopted by other initiatives\(^2, 3\). Web applications such as genome browsers rely heavily on JavaScript, a popular language for processing and rendering client-side information in a web browser. Despite their widespread use in bioinformatics, biological web applications are usually implemented with no standard reutilisation guidelines in mind, hence BioJS was developed\(^4\).

BioJS is an open source JavaScript library of components for the visualisation of biological data on the web. Here we present wigExplorer, a standard, portable BioJS component designed to easily render wig data format files. wigExplorer can be integrated and controlled from other applications. To our knowledge, this is the first modular component to visualise wig data that complies with BioJS standards.

The wigExplorer component
wigExplorer is fully integrated in the BioJS project. It follows the standards set by the BioJS registry\(^5\), a centralised repository of BioJS components hosted at the European Bioinformatics Institute (EBI). Having wigExplorer in the BioJS registry is advantageous because it promotes i) easy discoverability for the component, ii) collaborative development with other members of the BioJS community and iii) reutilisation by third party applications. In the BioJS registry, component APIs are exposed, i.e., events and methods are defined and documented so that other BioJS components can interact with each other. By following these conventions, wigExplorer is able to interact with other components on the same web page, enriching the overall experience for the user.

The code below shows how to incorporate wigExplorer into a web application. Only three configuration elements are needed: the target HTML element in which the component will be rendered, the background colour of the component, and the file path containing the wig data.

```javascript
var instance = new Biojs.wigExplorer({
  target: "YourOwnDivId",
  selectionBackgroundColor: '<background-colour>',
  dataSet: "<path-to-file>
});
```

wigExplorer uses D3.js, the data-driven documents JavaScript library\(^6\), to generate graphical representations from wig data. D3.js handles the manipulation of the data documents, the reading of wig data as text format and their conversion to an area chart format. To control the visual aspect of the wig data, wigExplorer contains simple controls for zooming and panning.

Application
wigExplorer can be used to visualise genomic data in different ways. An application is shown in Figure 1, depicting single nucleotide polymorphism (SNP) density data from a genomic annotation in the tomato genome. Here chromosome 2 is zoomed in to show the genomic interval contained between position 3M and 47M. The SNP density data contained in the wig data file are presented as bins, where the Y axis indicates the number of SNPs contained in each bin. The screenshot shows a dramatic change in the density of SNPs at the 24M bin mark of the chromosome, suggesting a potential boundary for an introgression segment introduced from a closely related tomato species. Other potential applications of wigExplorer may involve the visualisation of gene expression and alignment data.

We are aware that third party browsers are also using wigExplorer. A screenshot of the TGAC Browser\(^7\) is shown in Figure 2 using wigExplorer.

![Figure 1. wigExplorer view of tomato variety Heinz chromosome 2](image1.png)

**Figure 1.** wigExplorer view of tomato variety Heinz chromosome 2. The top controls are designed to scroll sideways. Peaks show SNP density of 1kb size bins. A change of SNP density can be observed around the 29M mark, with a slightly greater density of SNPs on the right, indicative of a potential introgression segment from another related species.
wigExplorer to depict *Myzus* spp. scaffold 1 zoomed in between regions 714K and 727K. Here strand-specific RNA-Seq paired-end read coverage is shown as a wig track. The track below shows a closely related annotated species gene set for comparison. This comparison suggests a potential gene extension in both forward and reverse orientation.

**Conclusions**
The wigExplorer component provides a platform to visualise biological data in wig format. wigExplorer can be easily integrated with other web components or extended to provide new functionality. We expect this component to be particularly useful for visualisation in a variety of data types such as SNP density, alignments and gene expression. Like any other BioJS component, wigExplorer requires little technical knowledge for its utilisation.

**Software availability**
Zenodo: wigExplorer, a BioJS component to visualise wig data, doi: 10.5281/zenodo.7721
GitHub: BioJS, http://github.com/biojs/biojs;

**Author contributions**
AT and RJ developed the code for wigExplorer. MC and GK created the user cases for Figure 1 and Figure 2 respectively. AT, RD, MC and GK wrote the paper.

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

**Grant information**
AT, GK MC and RD were supported by the UK Biotechnology and Biological Sciences Research Council (BBSRC) National Capability Grant (BB/J010375/1) at TGAC.

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

**Acknowledgements**
We are grateful to all BioJS developers who have contributed their work under an open source license. We are grateful to David Swarbreck at TGAC for his advice on the data shown in Figure 2.

**References**
1. Kent WJ, Sugnet CW, Furey TS, et al.: *The human genome browser at UCSC.* Genome Res. 2002; 12(6): 996–1006. PubMed Abstract | Publisher Full Text | Free Full Text
2. Donlin MJ: *Using the Generic Genome Browser (GBrowse).* Curr Protoc Bioinformatics. John Wiley and Sons, Inc., 2009. PubMed Abstract | Publisher Full Text
3. Skinner ME, Uzilov AV, Stein LD, et al.: *JBrowse: A next-generation genome browser.* Genome Res. 2009; 19(9): 1630–1638. PubMed Abstract | Publisher Full Text | Free Full Text
4. 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
5. BioJS: registry. http://www.ebi.ac.uk/Tools/biojs/registry/, 2013. Reference Source
6. D3.js data-driven documents. http://d3js.org, 2012. Reference Source
7. Thanki AS, Bian X, Davey RP, et al.: *TGAC Browser: visualisation solutions for big data in the genomic era.* 2013. Reference Source
8. Thanki AS, Jiminez RC, Kadakkottil G, et al.: *wigExplorer, a BioJS component to visualise wig data.* Zenodo. 2014. Data Source
Open Peer Review

Current Peer Review Status: ✗ ❓

Version 1

Reviewer Report 07 March 2014

https://doi.org/10.5256/f1000research.3732.r3697

© 2014 Lord P. 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.

Phil Lord
School of Computing Science, University of Newcastle, Newcastle, UK

I have tried this piece of software and it does appear to do what it says and running the software was not too difficult. The link to the software on Zenodo in the Software Availability section (Zenodo: wigExplorer, a BioJS component to visualise wig data, doi:10.5281/zenodo.77218) seems a bit pointless as the download bundle requires a large number of dependencies which are not included and for which paths have been hardcoded to local machine locations. The authors should also include either a tag or a commit hash so that the version of the software in the download bundle can be identified in GitHub.

I ran the software from a GitHub checkout - here all the paths given in the documentation assume a specific configuration of directories (which is different from that of the GitHub repository). None of this is a show stopper, but it does make me rather doubt the validity of the statement: "Like any other BioJS component, wigExplorer requires little technical knowledge for its utilisation."

The article is, essentially, an advert with relatively little technical content. This is appropriate; there is little point including, for example, precise instructions for the use of wigExplorer as this article will be checkpointed and the code will become out of date. I would however have liked a bit more information on the expected performance of the software: for instance, how big a chromosome can it visualize? How manipulable is the visualisation?

The writing in the article is okay, but there are a few parts which could do with improvement. For example, in the first paragraph of the Introduction section, the statement "biological web applications are usually implemented with no standard reutilisation guidelines in mind, hence BioJS was developed" I am not at all sure what "reutilisation guidelines" means with respect to code base. Do the authors just mean a defined and documented API? In which case, they should probably say this.

There are also a couple of places which could do with simplification. In the second paragraph of the Introduction section: "To our knowledge, this is the first modular component to visualise wig data"
that complies with BioJS standards." This statement could be seen as a caveat, and I think the authors could risk saying "This is the only BioJS component to show wig data". Also in the second paragraph of the Application section the statement "We are aware that third party browsers are also using wigExplorer." could perhaps be simplified to "Third party browsers are also using wigExplorer".

**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.

---

**Author Response 23 May 2014**

Anil Thanki, The Genome Analysis Centre, UK

We would like to thank Phil for taking the time to review the manuscript.

Please find below our responses to your comments

1) The link to the software on Zenodo in the Software Availability section ([Zenodo: wigExplorer, a BioJS component to visualise wig data, doi:10.5281/zenodo.77218](http://dx.doi.org/10.5281/zenodo.77218)) seems a bit pointless as the download bundle requires a large number of dependencies which are not included and for which paths have been hardcoded to local machine locations. The authors should also include either a tag or a commit hash so that the version of the software in the download bundle can be identified in GitHub.

Answer: We have uploaded a new version of the code to Zenodo with dependencies provided ([http://dx.doi.org/10.5281/zenodo.8516](http://dx.doi.org/10.5281/zenodo.8516)). This new version is also included in BioJS Release 1.0 ([https://github.com/biojs/biojs/releases/tag/v1.0](https://github.com/biojs/biojs/releases/tag/v1.0))

2) It does make me rather doubt the validity of the statement: "Like any other BioJS component, wigExplorer requires little technical knowledge for its utilisation."

Answer: We understand that there are a number of required dependencies for this component, and not including these in the component's distribution made it harder to install and utilise. As above, we have included these dependencies as well as sample datasets, so simply unzipping the distribution is sufficient to install and start using the component.

3) The article is, essentially, an advert with relatively little technical content. This is appropriate; there is little point including, for example, precise instructions for the use of wigExplorer as this article will be checkpointed and the code will become out of date. I would however have liked a bit more information on the expected performance of the software: for instance, how big a chromosome can it visualize? How manipulable is the visualisation?

Answer: *F1000Research* requires a working copy of the code when article get published (for
archival purposes) so, we have uploaded current version of code to zenodo (https://zenodo.org/record/8516#.U334LhvdV0s) so this copy of code will be same though main code can be updated. So this article will always point to the version of the code that it describes, which is stored in zenodo. And on the performance issue, in this case size of the file matters rather than size of genomic region. Because size of data depends on depth of the genomic information, though we tried to make an effort to make it clear in manuscript. In first paragraph of wigExplorer component section, “Wig files contains minimalistic information of genomic data, wigExplorer can handle large genomic region like a chromosome (tested with a single file containing 12 chromosome with average length of 60 Mb), but it depends on depth of data rather than length of genomic region."

4) The writing in the article is okay, but there are a few parts, which could do with improvement. For example, in the first paragraph of the Introduction section, the statement "biological web applications are usually implemented with no standard reutilisation guidelines in mind, hence BioJS was developed" I am not at all sure what "reutilisation guidelines" means with respect to code base. Do the authors just mean a defined and documented API? In which case, they should probably say this.

Answer: Here by using the word "reutilisation guidelines", we mean that BioJS components can be easily implemented and can interact with each other with standard API. We have added a statement explaining this in the last line of the first paragraph of introduction that “BioJS code contains standard guidelines that how to use components and how API can be implemented to interact with other components."

5) There are also a couple of places, which could do with simplification. In the second paragraph of the Introduction section: "To our knowledge, this is the first modular component to visualise wig data that complies with BioJS standards." This statement could be seen as a caveat, and I think the authors could risk saying "This is the only BioJS component to show wig data". Also in the second paragraph of the Application section the statement "We are aware that third party browsers are also using wigExplorer." could perhaps be simplified to "Third party browsers are also using wigExplorer". 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, however I have significant reservations, as outlined above.

Answer: We have reformatted the first statement to “Currently, this is the only BioJS component to show wig data” and we have changed the second statement as reviewer suggested, “Third party browsers are also using wigExplorer”

**Competing Interests:** No competing interests were disclosed.
Robert Buels
JBrowse, University of California, Berkeley, CA, USA

The manuscript is well written, but it does not accurately describe the software in question.

I installed the software on my machine, and evaluated its behavior under several scenarios that I think would be commonly encountered by people using it. The major issues I found with the software are:

1. **wigExplorer** does not provide any useful error messages when it fails to fetch, or fails to parse, the file specified in its `dataSet` option. This can be a major issue for prospective users (i.e. installers) of the software.

2. Also, when I installed it and ran it via the included TestwigExplorer2.html, the "top controls ... designed to scroll sideways" described in the manuscript were not shown. To rule out any problems with the way I installed it (caused by missing stylesheets or the like), I inspected the browser's (Google Chrome 32) document object model (DOM) using its built-in development tools, and verified that the controls were not even being placed in the DOM by the code itself. I didn't debug any further.

3. Most importantly, and I think this is a show stopper: wigExplorer does not actually support the Wiggle format as set forth by UCSC at [http://genome.ucsc.edu/goldenPath/help/wiggle.html](http://genome.ucsc.edu/goldenPath/help/wiggle.html). It supports a limited tab-separated value format, which bears some resemblance to BedGraph format, but is not BedGraph either. So: it is named for a standard data format that it does not support, and the data format that it actually supports is not a standard format.

Unfortunately, I cannot approve this manuscript as it is currently written. It does not accurately describe the wigExplorer software.

**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 state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

---

**Reader Comment 26 Feb 2014**

**Anil Thanki**, The Genome Analysis Centre, UK

Hi Robert,

I thank you for your useful comments, and I fully understand your concerns about the code and the performance of the tool. I have already started working on fixes regarding your comments, and happy to say that the first two of your three comments have already been fixed. I am now working on the third point and hope to have a solution as soon as possible. Once I do have one I will submit a revised version of the new code and manuscript.
Thanks a lot again,

Anil

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

Author Response 28 May 2014

**Anil Thanki**, The Genome Analysis Centre, UK

Dear Robert

We would like to thank you for taking the time to review the manuscript.

Please find below our responses to your comments.

1) *wigExplorer* does not provide any useful error messages when it fails to fetch, or fails to parse, the file specified in its `dataSet` option. This can be a major issue for prospective users (i.e. installers) of the software.

   **Answer:** After considering this comment we have added more informative error messages, depending on the errors that can occur while running the component, e.g. “Unknown format detected”, “No data for selected reference”, “File not found”, etc. This will help make any installation problems clearer to users during installation.

2) Also, when I installed it and ran it via the included TestwigExplorer2.html, the "top controls ... designed to scroll sideways" described in the manuscript were not shown. To rule out any problems with the way I installed it (caused by missing stylesheets or the like), I inspected the browser's (Google Chrome 32) document object model (DOM) using its built-in development tools, and verified that the controls were not even being placed in the DOM by the code itself. I didn't debug any further.

   **Answer:** Outside of the BioJS registry environment, the component's CSS behaviour changes, which resulted in disappearing controls. We have reformatted the CSS for this component and this is now fixed.

3) Most importantly, and I think this is a showstopper: *wigExplorer* does not actually support the Wiggle format as set forth by UCSC at [http://genome.ucsc.edu/goldenPath/help/wiggle.html](http://genome.ucsc.edu/goldenPath/help/wiggle.html). It supports a limited tab-separated value format, which bears some resemblance to BedGraph format, but is not BedGraph either. So: it is named for a standard data format that it does not support, and the data format that it actually supports is not a standard format.

   **Answer:** We thanks Robert t raise this concern and we have rewritten the code and now the component fully supports the standard fixedStep and variableStep wig file format (as set out by UCSC) with span information and multiple references in one file if available.
**Competing Interests:** No competing interests were disclosed.

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