CrossMark
click for updates

SOFTWARE TOOL ARTICLE

REVISED *wigExplorer*, a BioJS component to visualise wig data [version 3; referees: 1 approved, 2 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]Earlham Institute, Norwich Research Park, Norwich, NR4 7UH, UK
[2]European Bioinformatics Institute, Hinxton, Cambridge, CB10 1SD, UK

v3 **First published:** 13 Feb 2014, **3**:53 (doi: 10.12688/f1000research.3-53.v1)
Second version: 30 May 2014, **3**:53 (doi: 10.12688/f1000research.3-53.v2)
Latest published: 09 Aug 2016, **3**:53 (doi: 10.12688/f1000research.3-53.v3)

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://biojs.io/d/biojs-vis-wigexplorer; http://dx.doi.org/10.5281/zenodo.8516

BIOJS

This article is included in the BioJS channel.

EMBL-EBI

This article is included in the EMBL-EBI channel.

Open Peer Review

Referee Status: ✗ ✓ ? ?

	Invited Referees 1	2	3	4
REVISED version 3 published 09 Aug 2016				? report
REVISED version 2 published 30 May 2014		✓ report	? report	
version 1 published 13 Feb 2014	✗ report	? report		

1 **Robert Buels**, University of California, Berkeley USA

2 **Phil Lord**, University of Newcastle UK

3 **Stefan Thomas Lang**, Lund University Sweden

4 **Chunlei Wu**, Scripps Research Institute USA, **Jiwen Xin**, The Scripps Research Institute USA

Discuss this article

Comments (0)

Corresponding author: Anil S. Thanki (Anil.Thanki@earlham.ac.uk)

How to cite this article: Thanki AS, Jimenez RC, Kaithakottil GG *et al.* ***wigExplorer*, a BioJS component to visualise wig data [version 3; referees: 1 approved, 2 approved with reservations, 1 not approved]** *F1000Research* 2016, **3**:53 (doi: 10.12688/f1000research.3-53.v3)

Grant information: AT, GK and RD were supported by the UK Biotechnology and Biological Sciences Research Council (BBSRC) National Capability Grant (BB/J010375/1) at Earlham Institute.
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**:53 (doi: 10.12688/f1000research.3-53.v1)

REVISED **Amendments from Version 2**

In the new version we have updated the availability sources, and some reference details based on the referees' comments.

See referee reports

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 code contains proper guidelines on how to use the components and how the API can be implemented to interact with other components.

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. Wig files contain minimalistic information of genomic data *wigExplorer* and can handle a large genomic region such as a chromosome (tested with a single file containing 12 chromosome with average length of 60 Mb), but this depends on the richness of the data rather than the length of the genomic region.

```
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. On the top right side, *wigExplorer* contains a dropdown to toogle between different references from the wig file. To control the visual aspect of the wig data, *wigExplorer* contains simple controls for zooming and panning. It is also possible to zoom and pan using provided API.

```
instance._updateDraw(start, end)
```

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 2.5M and 47.5M. 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 just after 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.

Third party browsers are also using *wigExplorer*. A screenshot of the TGAC Browser[7] is shown in Figure 2 using *wigExplorer* to depict *Myzus* spp. scaffold 1 zoomed in between regions 714K and 727K. Here strandspecific 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

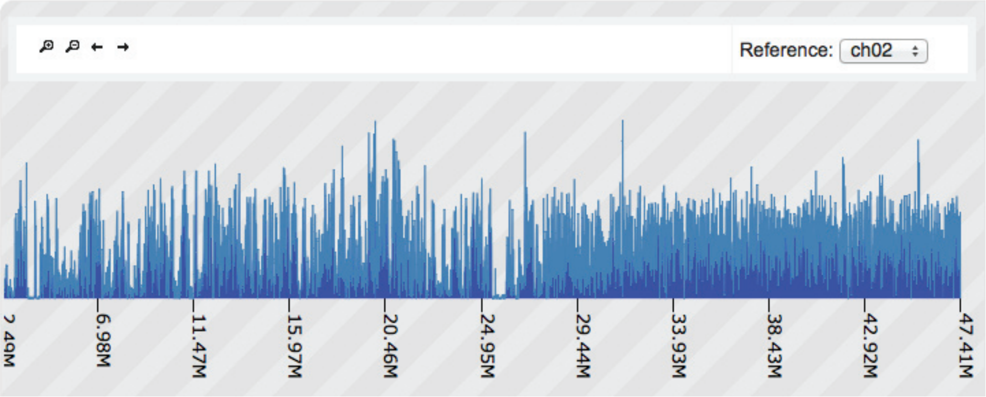

Figure 1. *wigExplorer* view of tomato variety Heinz chromosome 2. The top controls are designed to toogle between different references as well as zoom and pan. Peaks show SNP density of 1kb size bins. A change of SNP density can be observed around the 24M mark, with a slightly greater density of SNPs on the right, indicative of a potential introgression segment from another related species.

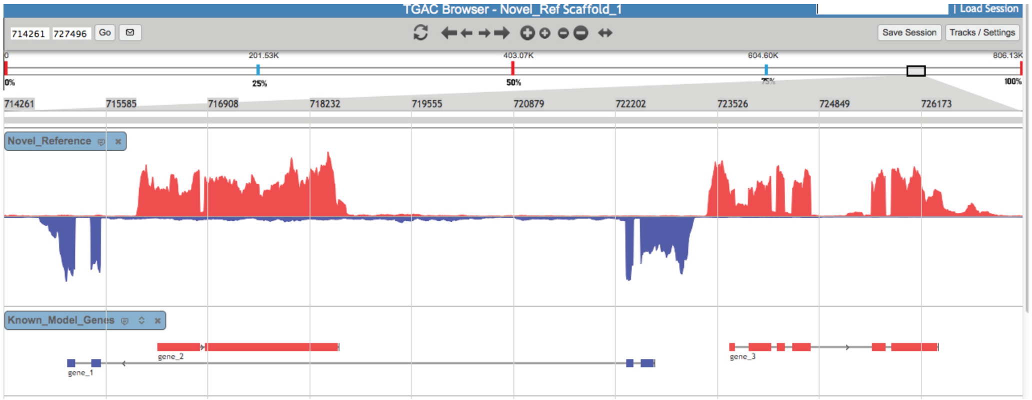

Figure 2. An example of *wigExplorer* integration using the TGAC Browser. The *wigExplorer* track shows read coverage in *Myzus* spp. for scaffold 1. Forward and backward strands are depicted in red and blue respectively. Evidence genes from a closely related species are displayed in the track below.

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_v2, doi: 10.5281/zenodo.8516[8]

GitHub: BioJS, http://github.com/biojs/biojs/releases/tag/v1.0;

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 and RD were supported by the UK Biotechnology and Biological Sciences Research Council (BBSRC) National Capability Grant (BB/J010375/1) at Earlham Institute.

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 thankful to David Swarbreck at Earlham Institute 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://biojs.io, 2016.
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.** 2016.
Reference Source

8. Thanki AS, Jiminez RC, Kaithakottil GK, *et al.*: **wigexplorer, a biojs component to visualise wig data_v2.** *Zenodo.* 2014.
Data Source

Open Peer Review

Current Referee Status: ☒ ☑ ? ?

Version 3

Referee Report 05 October 2016

doi:10.5256/f1000research.10145.r16472

?

Chunlei Wu[1], **Jiwen Xin**[2]

[1] Scripps Research Institute, La Jolla, CA, USA

[2] The Scripps Research Institute, La Jolla, CA, USA

This manuscript decribed a useful BioJS component for visualizing genomic tracks from a wig-formatted file. There are some major concerns raised by the reviewer:

- Demo code show “Unknown format detected” error. When clicking the demo code from the provided links to JSBin, Codepen and Plunker, all three have the same errors. This is also confirmed the same by running the demo code locally. But it can be fixed by changing the dataSet to:

  dataSet: "./wigExplorerDataSet5.txt"
  (was dataSet:
  "https://cdn.rawgit.com/anilthanki/biojs-vis-wigexplorer/master/snippets/data/wigExplorerDataSet5.

  Even thought, there still are some javascript errors displayed on the browser console (using Google Chrome 54) as described below.

- The website http://biojs.io/d/biojs-vis-wigexplorer shows two javascript errors, even though the rendering of the widget looks fine. I confirmed that these errors are from the wigexplorer itself, as I can see the same errors in the console by running the demo code locally. This is the screenshot of the javascript errors:

  http://imgur.com/a/sfWkK

- After several previous reviewers requested the actual code example for integrating WigExplorer into third-party applications like TGAC browser, you still did not provide much details of it.

  The styling of the tracks in TGAC Browser screenshot does not look similar to the demo example, which does raise the concerns about how the integration was actually done in a real application, and how flexible a developer can control the look and feel of the application. A real code example should address those concerns.

- I also tried the demo code locally with two example wig files downloaded from NCI website: WashU (74MB) and Baylor (79MB). The WashU example file does load, but the track rendering is very slow, and the track control buttons (like zoom and move) are pretty much unusable. The Baylor example file does not load, also without any error message. These two files could be out of the file size range for reasonable rendering in this widget, a proper error msg should be displayed ideally.

- A minor suggestion, although the project website is provided under "Software availability" section, it would be helpful to provide the direct links to the demo code at the relevant text.

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

Competing Interests: No competing interests were disclosed.

Version 2

Referee Report 07 June 2016

doi:10.5256/f1000research.4634.r14186

Stefan Thomas Lang
Department of Molecular Hematology, Lund University, Lund, Sweden

I am no JavaScript developer myself and I have not tested the described functionality, but I already can think of applications in my own work if the package is able to not only show predefined chromosomes in a wig file, but subsets in a chromosome too.

I will state all my issues like this first major one:

1. The link to your real example page (http://biojs.io/d/biojs-vis-wigexplorer) can only be found by reading your article, following to the rather unspecific GitHub link to the http://biojs.io web page. There one need to search for your wigexplorer - that is not acceptable! Make this link more prominent in the article and refer to the GitHub page as the main biojs.io GitHub page.

2. I can not approve this article, if your package can not display any user defined region in a wig file. After having checked the functionality on the hidden demo page I get the impression that this javascript package does ONLY visualize whole chromosomes from a wig file. The user can not change the visualized region by e.g. interactively zooming in or at least stating a displayed area in the javascript function call? But if your package can visualize a user defined area in a wig file this functionality needs to be highlighted in the article.

In addition I have several minor 'complains':

1. Can the package be used stand alone? If yes, please extend your code example to a minimal, working html page. It would be perfect to have the javascript library loaded from your own web server together with the example data. Much like on the hidden demo page, but as plain html plainly stated in the article.

2. You claim that your tool easily integrats into the third party TGAC Browser. Can you provide the example html there, too?

I hope I could add some additional points of view to your work and give a kind of positive feedback.

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.

Competing Interests: No competing interests were disclosed.

Author Response 03 Aug 2016

Anil Thanki, The Genome Analysis Centre, UK

Dear Stefan

We would like to thank you for reviewing the manuscript and making some positive comments that could be used for your application.

Please find below our responses to your comments.

1) The link to your real example page (http://biojs.io/d/biojs-vis-wigexplorer) can only be found by reading your article, following to the rather unspecific GitHub link to the http://biojs.io web page. There one need to search for your wigExplorer - that is not acceptable! Make this link more prominent in the article and refer to the GitHub page as the main biojs.io GitHub page.

Answer: We can understand your concern, as this component was developed as a part of BioJS 1 and now BioJS is evolved to version 2, so some links are broken or redirected. We will update links in the article according. Link : http://www.biojs.io/d/biojs-vis-wigexplorer

2) I cannot approve this article, if your package cannot display any user defined region in a wig file. After having checked the functionality on the hidden demo page I get the impression that this JavaScript package does ONLY visualize whole chromosomes from a wig file. The user cannot change the visualized region by e.g. interactively zooming in or at least stating a displayed area in the JavaScript function call? But if your package can visualize a user defined area in a wig file this functionality needs to be highlighted in the article.

Answer: I can understand your issue here about visualising user defined region, wigExplorer was already had a capability of allowing user to move around and zoom in and out for visualising region of interest, but due to compatibility issue with BioJS2 buttons were not visible. Now we had fixed the issue and you will notice some controls on top left corner which gives user flexibility to visualise any region. It is also possible to visualise user defined area using API provided.

3) Can the package be used stand alone? If yes, please extend your code example to a minimal, working html page. It would be perfect to have the javascript library loaded from your own web server together with the example data. Much like on the hidden demo page, but as plain html plainly stated in the article.

Answer: Yes, this package has been designed to work standalone, I have set up some example on my personal GitHub page to prove that. Here is the link to stand alone example for wigExplorer: http://anilthanki.github.io/tools/biojs/wigExplorer/

4) You claim that your tool easily integrates into the third party TGAC Browser. Can you provide the example html there, too?

Answer: Yes, we believe that it's easy to integrate into any third party application and TGAC Browser uses wigExplorer to visualise users' wig data. Here is an example explaining procedure to visualise your own wig data using TGAC Browser.
Example:

1. Go to link http://browser.tgac.ac.uk/tgac-browser-demo/index.jsp?query=chr1&coord=4&&from=0&&to=5
2. Click on the upload icon on the right (before 'Save Session')
3. There is a link to example wig file, to save and upload to see example
4. Upload the same file to the 'Drop files here' region and close the upload box
5. You should see the uploaded wig file displayed with prefix of the filename you uploaded.

Regards,

Anil Thanki

Competing Interests: No competing interests were disclosed.

Referee Report 01 March 2016

doi:10.5256/f1000research.4634.r4977

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

The authors have provided answers to all my comments.

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.

Version 1

Referee Report 07 March 2014

doi:10.5256/f1000research.3732.r3697

? **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*".

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.

Competing Interests: No competing interests were disclosed.

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) 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). This new version is also included in BioJS Release 1.0 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#.U334LlhdV0s) 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.

Referee Report 25 February 2014

doi:10.5256/f1000research.3732.r3699

☒ **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. 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.

I have read this submission. I 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.

Competing Interests: No competing interests were disclosed.

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. 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.