# biojs-io-biom, a BioJS component for handling data in Biological Observation Matrix (BIOM) format

**Journal Article**

**Author(s):**
Ankenbrand, Markus J.; Terhoeven, Niklas (ID); Hohlfeld, Sonja; Förster, Frank; Keller, Alexander

Check for updates

SOFTWARE TOOL ARTICLE

# REVISED biojs-io-biom, a BioJS component for handling data in Biological Observation Matrix (BIOM) format [version 2; peer review: 2 approved, 1 approved with reservations]

Markus J. Ankenbrand[1], Niklas Terhoeven[2,3], Sonja Hohlfeld[1], Frank Förster[3,4], Alexander Keller[1]

[1]Department of Animal Ecology and Tropical Biology (Zoology III), University of Würzburg, Würzburg, Germany
[2]Department of Plant Physiology and Biophysics (Botany I), University of Würzburg, Würzburg, Germany
[3]Center for Computational and Theoretical Biology (CCTB), University of Würzburg, Würzburg, Germany
[4]Department of Bioinformatics, University of Würzburg, Würzburg, Germany

## Abstract

The Biological Observation Matrix (BIOM) format is widely used to store data from high-throughput studies. It aims at increasing interoperability of bioinformatic tools that process this data. However, due to multiple versions and implementation details, working with this format can be tricky. Currently, libraries in Python, R and Perl are available, whilst such for JavaScript are lacking. Here, we present a BioJS component for parsing BIOM data in all format versions. It supports import, modification, and export via a unified interface. This module aims to facilitate the development of web applications that use BIOM data. Finally, we demonstrate its usefulness by two applications that already use this component.

**Availability:** https://github.com/molbiodiv/biojs-io-biom,
https://dx.doi.org/10.5281/zenodo.218277

**Keywords**
biom-format , ecology , meta-genomics , biojs , parser , meta-barcoding

This article is included in the BioJS collection.

## Open Peer Review

**Reviewer Status** ? ✔ ✔

| | Invited Reviewers | | |
|---|---|---|---|
| | **1** | **2** | **3** |
| REVISED **version 2** published 09 Jan 2017 | | ✔ report | ✔ report |
| | | ↑ | ↑ |
| **version 1** published 20 Sep 2016 | ? report | ? report | ? report |

1 **Daniel McDonald**, University of California, San Diego, La Jolla, USA

**Evan Bolyen**, Northern Arizona University, Flagstaff, USA

2 **Holly M. Bik** (iD), University of California Riverside, Riverside, USA

3 **Joseph Nathaniel Paulson**, Harvard T.H. Chan School of Public Health, Boston, USA

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

**Corresponding author:** Markus J. Ankenbrand (markus.ankenbrand@uni-wuerzburg.de)

**How to cite this article:** Ankenbrand MJ, Terhoeven N, Hohlfeld S *et al.* **biojs-io-biom, a BioJS component for handling data in Biological Observation Matrix (BIOM) format [version 2; peer review: 2 approved, 1 approved with reservations]** F1000Research 2017, **5**:2348 ( https://doi.org/10.12688/f1000research.9618.2)

**First published:** 20 Sep 2016, **5**:2348 (https://doi.org/10.12688/f1000research.9618.1)

## Introduction

In recent years, there has been an enormous increase in biological data available from high-throughput studies. Complications arise from the enlarged size of the resulting data tables. This is the case for transcriptomic and marker-gene community data, where the central matrix consists of counts for each observation (e.g. gene or taxon) in each sample, plus a second and third matrix for metadata of both taxa and samples, respectively.

Early on there have been efforts to define data formats that capture all relevant information for an experiment like the Minimum Information About a Microarray Experiment (MIAME) project[1]. In 2005 the Genomic Standards Consortium (GSC) formed with the mission of enabling genomic data integration, discovery and comparison through international community-driven standards[2]. The Biological Observation Matrix (BIOM) Format was developed to standardize the storage of observation counts together with all relevant metadata and it is a member project of the GSC[3]. One main purpose of the BIOM format is to enhance interoperability between different software suits. Many current leading tools in community ecology and metagenomics support the BIOM format, e.g. QIIME[4], MG-RAST[5], PICRUSt[6], phyloseq[7], VAMPS[8] and Phinch[9]. Additionally, libraries exist in Python[3], R[10] and Perl[11] to propagate the standardized use of the format.

Interactive visualization of biological data in a web browser is becoming more and more popular[12,13]. For the development of web applications that support BIOM data, a corresponding library is currently lacking and would be very useful, since several challenges arise when trying to handle BIOM data. While BIOM format version 1.0 builds on the JSON format and thus is natively supported by JavaScript, the more recent BIOM format version 2.1 uses HDF5 and can therefore not be handled natively in web browsers. Also the internal data storage can be either dense or sparse so applications have to handle both cases. Furthermore application developers need to be very careful when modifying BIOM data as changes that do not abide to the specification will break interoperability with other tools. Here we present biojs-io-biom, a JavaScript module that provides a unified interface to read, modify, and write BIOM data. It can be readily used as a library by applications that need to handle BIOM data for import or export directly in the browser. To demonstrate the utility of our module it has been used to implement a simple user interface for the biom-conversion-server[14]. Additionally, the popular BIOM visualization tool Phinch[9] has been extended with new features, in particular support for BIOM version 2.1 by integrating biojs-io-biom[15].

### The biojs-io-biom component

The biojs-io-biom library can be used to create new objects (called `Biom` objects for brevity) by either loading file content directly via the static `parse` function or by initialization with a JSON object:

```
var biom = new Biom({
    id: 'My Biom',
    matrix_type: 'dense',
    shape: [2,2],
    rows: [
        {id: 'row1', metadata: {}},
        {id: 'row2', metadata: {}}
    ],
    columns: [
        {id: 'col1', metadata: {}},
        {id: 'col2', metadata: {}}
    ],
    data: [
        [0,1],
        [2,3]
    ]
});
```

The data is checked for integrity and compliance with the BIOM specification. Missing fields are created with default content. All operations that set attributes of the `Biom` object with the dot notation are also checked and prompt an error if they are not allowed.

```
var biom = new Biom({});
biom.id = [];
// Will throw a TypeError as id has to be a
string or null
```

Beside checking and maintaining integrity the biojs-io-biom library implements convenience functions. This includes getter and setter for metadata as well as data accessor functions that are agnostic to internal representation (dense or sparse). But one of the main features of this library is the capability of handling BIOM data in both versions 1.0 and 2.1 by interfacing with the biom-conversion-server[14]. Handling of BIOM version 2.1 in JavaScript directly is not possible due to its HDF5 binary format. The only reference implementation of the format is in C and trying to transpile the library to JavaScript using emscripten[16] failed due to strong reliance on file operations (see discussions in[17,18]). Using the conversion server allows developers to use BIOM of both versions transparently. `Biom` objects also expose the function `write` which exports it as version 1.0 or version 2.1. In contrast to the existing `biom_convert` module for the Galaxy platform which has a rich set of options the biom-conversion-server exhibits its functionality both via an API and a simple user interface that does not need any kind of setup or login[19,20].

## Application

To demonstrate the utility of this module it has been used to implement a user interface for the biom-conversion-server[14]. Besides providing an API it is now also possible to upload files using a file dialog. The uploaded file is checked using our module and converted to version 1.0 on the fly if necessary. It can then be downloaded in both version 1.0 and 2.1. As most of the functionality is provided by the biojs-io-biom module the whole interface is simply implemented with a few additional lines of code.

As a second example the Phinch framework[9] has been enhanced to allow BIOM version 2.1. Phinch visualizes the content of BIOM files using a variety of interactive plots. However due to the difficulties of handling HDF5 data only BIOM version 1.0 is supported. This is unfortunate as most tools nowadays return BIOM version 2.1 (e.g. QIIME from version 1.9,1[4] and Qiita[21]). It is possible to convert from version 2.1 to version 1.0 without loss of information but that requires an extra step using the command line. By including our biojs-io-biom module and the biom-conversion-server into Phinch it was possible to add support for BIOM version 2.1 along with some other improvements[15].

As the biojs-io-biom module resolves the import and export challenges, one of the next steps is the development of a further BioJS module to present BIOM data as a set of data tables. In order to do that for large datasets sophisticated, accessor functions capitalizing on the sparse data representation have to be implemented.

A drawback of the internal storage of BIOM version 1.0 is that it suffers of those shortcomings that are solved in version 2.1, specifically efficient handling of huge datasets. However even with a more efficient data storage huge amounts of data will still cause problems with current web browsers. Therefore, we plan on extending the biom-conversion-server with a light communication API that allows a client to request only the subsets of the full data set that it requires.

## Conclusion

The module biojs-io-biom was developed to enhance the import and export of BIOM data into JavaScript. Its utility and versatility has been demonstrated in two example applications. It is implemented using latest web technologies, well tested and well documented. It provides a unified interface and abstracts from details like version

or internal data representation. Therefore, it will facilitate the development of web applications that rely on the BIOM format.

## Software availability

### biojs-io-biom

Latest source code https://github.com/molbiodiv/biojs-io-biom

Archived source code as at the time of publication https://zenodo.org/record/218277

License MIT

### biom-conversion-server

Latest source code https://github.com/molbiodiv/biom-conversion-server

Archived source code as at the time of publication https://zenodo.org/record/218396

Public instance https://biomcs.iimog.org

License MIT

---

## References

1. Brazma A, Hingamp P, Quackenbush J, *et al.*: **Minimum information about a microarray experiment (MIAME)-toward standards for microarray data.** *Nat Genet.* 2001; **29**(4): 365–371.
   **PubMed Abstract** | **Publisher Full Text**

2. Field D, Amaral-Zettler L, Cochrane G, *et al.*: **The Genomic Standards Consortium.** *PLoS Biol.* 2011; **9**(6): e1001088.
   **PubMed Abstract** | **Publisher Full Text**

3. McDonald D, Clemente JC, Kuczynski J, *et al.*: **The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome.** *Gigascience.* 2012; **1**(1): 7.
   **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

4. Caporaso JG, Kuczynski J, Stombaugh J, *et al.*: **QIIME allows analysis of high-throughput community sequencing data.** *Nat Methods.* 2010; **7**(5): 335–336.
   **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

5. Meyer F, Paarmann D, D'Souza M, *et al.*: **The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes.** *BMC Bioinformatics.* 2008; **9**: 386.
   **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

6. Langille MG, Zaneveld J, Caporaso JG, *et al.*: **Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences.** *Nat Biotechnol.* 2013; **31**(9): 814–821.
   **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

7.  McMurdie PJ, Holmes S: **Phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data.** *PLoS One.* 2013; **8**(4): e61217.
    **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

8.  Huse SM, Mark Welch DB, Voorhis A, *et al.*: **VAMPS: a website for visualization and analysis of microbial population structures.** *BMC Bioinformatics.* 2014; **15**: 41.
    **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

9.  Bik HM; Pitch Interactive: **Phinch: An interactive, exploratory data visualization framework for –Omic datasets.** *bioRxiv.* 2014; 009944.
    **Publisher Full Text**

10. McMurdie PJ, Paulson JN: **biomformat: An interface package for the BIOM file format.** R/Bioconductor package version 1.0.0. 2015.

11. Angly FE, Fields CJ, Tyson GW: **The Bio-Community Perl toolkit for microbial ecology.** *Bioinformatics.* 2014; **30**(13): 1926–1927.
    **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

12. Corpas M, Jimenez R, Carbon SJ, *et al.*: **BioJS: an open source standard for biological visualisation - its status in 2014 [version 1; referees: 2 approved].** *F1000Res.* 2014; **3**: 55.
    **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

13. Corpas M: **The BioJS article collection of open source components for biological data visualisation [version 1; referees: not peer reviewed].** *F1000Res.* 2014; **3**: 56.
    **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

14. Ankenbrand MJ: **molbiodiv/biom-conversion-server: Version 1.0.2**. 2016.
    **Publisher Full Text**

15. **Pull request #67· PitchInteractiveInc/Phinch.** preview version online at **https://blackbird.iimog.org**. Accessed: 2016-12-22.
    **Reference Source**

16. Kripken/emscripten: **Emscripten: An LLVM-to-JavaScript Compiler.** Accessed: 2016-09-08.
    **Reference Source**

17. **Biom javascript module· Issue #699· biocore/biom-format.** Accessed: 2016-09-08.
    **Reference Source**

18. **hdf5 javascript in a webbrowser· Issue #29· HDF-NI/hdf5.node.** Accessed: 2016-09-08.
    **Reference Source**

19. Afgan E, Baker D, van den Beek M, *et al.*: **The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update.** *Nucleic Acids Res.* 2016; **44**(W1): W3–W10.
    **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**

20. **biom convert galaxy module.** Accessed: 2016-12-15.
    **Reference Source**

21. **Qiita.** Accessed: 2016-09-08.
    **Reference Source**

# Open Peer Review

## Current Peer Review Status:  ?  ✓  ✓

**Version 2**

Reviewer Report 19 April 2018

✓   **Holly M. Bik**  iD
Department of Nematology, University of California, Riverside, Riverside, CA, USA

The authors have made significant changes and the revised manuscript is much improved over the original version. The focus on the biojs-io-biom tool provides a much better rationale and manuscript structure. I am satisfied with the author responses to my previous comments and now happy to give final approval to this manuscript.

*Competing Interests:* I am the Principal Investigator on the Phinch framework (http://phinch.org), and our codebase was used as a case study in this manuscript.

**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 09 January 2017

✓   **Joseph Nathaniel Paulson**
Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA, USA

The authors addressed my main concerns and I have noticed that the documentation is much better on the github page. Good job

*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.**

**?**

**Joseph Nathaniel Paulson**

Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA, USA

Ankenbrand *et al.* provide a javascript library to interact with the microbial consortia BIOM format version 1 class. As the authors note, a javascript library could be a great benefit to the community as many commonly used tools like QIIME and Mothur produce BIOM formatted objects. However, the article and software are missing a few key components for a fully positive review.

Major comments:

There is a historical context that Ankenbrand *et al.* miss in discussing biom-format and subsequently imply that the biom-format is more widely adopted than being field specific format. If the authors leave the introduction more general, then I would suggest they include more background on the history of high-throughput data storage and reproducibility in programmatic languages, perhaps starting with the Minimum Information About a Microarray Experiment - MIAME format [1] and exprSet classes developed in R about 15 years ago before the genomics standards consortium (formed in 2005), for which biom-format is a member.

The authors posit that the BIOM format version 2 / 2.1 that moved to HDF5 made it impossible for javascript libraries to manipulate it natively. We found a javascript library that "takes advantage of the compatibility of V8 and HDF5". Were the authors unable to build from this library to take advantage of the version 2 BIOM format? The BIOM version 2 / 2.1 formats were designed specifically to handle many of the shortcomings of the version 1 in terms of memory and design. It would be advantageous of the users to build from this if possible to at least read in the BIOM v2.1 HDF5 files.

In my own installation of the software, I keep getting error messages when I attempt to create a biom object, see here: http://tinyurl.com/f1000-review. If the reviewers could please clarify the installation guide on the github repo.

Minor comments:

The second sentence needs clarification. "Despite this increase, for many of these studies the general basic layout of the data is similar to traditional assessment after bioinformatical processing, yet

complications arise due to the increased size of the data tables."

The citation for the BIOM interface R package has been deprecated. The appropriate citation is: Paul J. McMurdie and Joseph N Paulson (2015). *biomformat: An interface package for the BIOM file format.* R/Bioconductor package version 1.0.0.[2].

**References**

1. Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, Aach J, Ansorge W, Ball CA, Causton HC, Gaasterland T, Glenisson P, Holstege FC, Kim IF, Markowitz V, Matese JC, Parkinson H, Robinson A, Sarkans U, Schulze-Kremer S, Stewart J, Taylor R, Vilo J, Vingron M: Minimum information about a microarray experiment (MIAME)-toward standards for microarray data.*Nat Genet*. 2001; **29** (4): 365-71 PubMed Abstract | Publisher Full Text
2. McMurdie PJ, Paulson JN: biomformat: An interface package for the BIOM file format. *R/Bioconductor package version 1.0.0*. 2015. Reference Source

*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 Dec 2016

**Markus J. Ankenbrand**, University of Würzburg, Würzberg, Germany

Thanks a lot for the thorough review and the good suggestions for improvement. Find our point by point answers below (original comments in bold):

**There is a historical context that Ankenbrand et al. miss in discussing biom-format and subsequently imply that the biom-format is more widely adopted than being field specific format. If the authors leave the introduction more general, then I would suggest they include more background on the history of high-throughput data storage and reproducibility in programmatic languages, perhaps starting with the Minimum Information About a Microarray Experiment - MIAME format 1 and exprSet classes developed in R about 15 years ago before the genomics standards consortium (formed in 2005), for which biom-format is a member.**
As suggested we extended the introduction to cover more of the historical context.

**The authors posit that the BIOM format version 2 / 2.1 that moved to HDF5 made it impossible for javascript libraries to manipulate it natively. We found a javascript library that "takes advantage of the compatibility of V8 and HDF5". Were the authors unable to build from this library to take advantage of the version 2 BIOM format? The BIOM version 2 / 2.1 formats were designed specifically to handle many of the shortcomings of the version 1 in terms of memory and design. It would be advantageous of the users to build from this if possible to at least read in the BIOM v2.1 HDF5 files.**
There is a fine distinction between JavaScript inside a browser and on a server (nodejs) that we previously did not make sufficiently clear in our manuscript. For the nodejs environment there is in fact a library that handles data in HDF5 format (https://github.com/HDF-NI/hdf5.node). As our library is supposed to work equally well in both environments we tried to port this library to the

browser. Unfortunately that proofed to be infeasible even after contacting the developers of the library (see https://github.com/HDF-NI/hdf5.node/issues/29). We adjusted the manuscript to make clear that HDF5 is not natively supported in the browser rather than in javascript in general. Further we added a section discussing the downside of being limited to JSON and plans to overcome that at the end of the Application section.

**In my own installation of the software, I keep getting error messages when I attempt to create a biom object, see here: http://tinyurl.com/f1000-review. If the reviewers could please clarify the installation guide on the github repo.**
Thanks for finding that issue. We fixed the bug creating your issue, added a minimum required version of nodejs and improved the documentation.

**The second sentence needs clarification. "Despite this increase, for many of these studies the general basic layout of the data is similar to traditional assessment after bioinformatical processing, yet complications arise due to the increased size of the data tables."**
Rephrased

**The citation for the BIOM interface R package has been deprecated. The appropriate citation is: Paul J. McMurdie and Joseph N Paulson (2015). biomformat: An interface package for the BIOM file format. R/Bioconductor package version 1.0.0.2.**
Fixed

*Competing Interests:* No competing interests were disclosed.

Reviewer Report 18 October 2016

https://doi.org/10.5256/f1000research.10362.r16436

? **Holly M. Bik** (iD)

Department of Nematology, University of California, Riverside, Riverside, CA, USA

This manuscript describes the biojs-io-biom toolkit, which includes a conversion library and server for re-formatting Biological Observation Matrix (BIOM) files between versions 1.x (JSON-formatted) and 2.x (HDF5-formatted).

The conversion library itself is extremely useful, since it will allow users to convert quickly between BIOM file formats without having to go back to the command line (e.g. QIIME) and easily reformat files for use in various applications.

I do not have the necessary javascript expertise to comment on the codebase and conversion server

backend, so I will offer some general comments on the practical applications outlined in the text:

Since this project is based on the Phinch framework, I find the "Blackbird" rebranding of the fork to be very problematic. The "Blackbird" instance is really just an updated release of the Phinch framework, with some bug fixes, added features, and implementation of the new BIOM conversion server. The rebranding/renaming is confusing for the end user (see comment by other peer reviewer below), and mistakenly implies a number of scenarios that are not accurate: 1) that the authors were involved in the original development of data visualization tools, 2) that the Blackbird rebranding and design changes were approved from by the original developers, and 3) the "Blackbird" project represents a significant expansion or retooling of the current Phinch framework. I'm fully aware that this is open source software and the authors are free to reuse and share the Phinch codebase, but I don't really see the utility of the "Blackbird" rebranding, and creating an additional web instance that mostly replicates the functionality of http://phinch.org will confuse end users.

Since the authors here are really community contributors to the original Phinch project, I would recommend eliminating the "Blackbird" rebranding of the project, and reverting back to Phinch branding (citing the framework release as Phinch v2.0). We will then initiate a pull request to update the bug fixes and integrate the new biojs-io-biom source code to be live on http://phinch.org  The visual layout for Phinch (name, logo and visualization layout) was thoughtfully constructed, and the new Blackbird logo and visual modifications will likely interfere with "brand recognition" that should be attributed to the original Phinch framework.

Once this pull request is initiated and completed, the "Application" manuscript text should be updated to reflect the live implementation of the conversion library on a v2.0 Phinch framework at phinch.org.

Other minor comments:
- Can you please provide details on how and where the "Blackbird" instance and biom-conversion-server are currently hosted (e.g. Amazon AWS)?

- Please list the public landing page for the applications mentioned in the text (in case users want to access these tools directly) - e.g. https://biomcs.iimog.org

- The biom-conversion-server does not appear to be backwards compatible (I could not upload and convert a BIOM 1.x file to 2.x format) - this one-way conversion functionality is should be clearly indicated in the first paragraph of the "Application" section. In addition, if users try to upload a BIOM 1.0 file they should be presented with an appropriate error message (I didn't see one - the tool just froze when I attempted to upload a BIOM 1.0 file).

- There are other BIOM conversion servers that exist, e.g. implementations within the Galaxy framework - see https://toolshed.g2.bx.psu.edu/repository/display_tool?repository_id=b3ae8ca9317b000e&render_ - these alternate tools should be mentioned in the text. How does the biom-conversion-server compare with (and potentially improve on) such Galaxy based tools?

***Competing Interests:*** I am the Principal Investigator on the Phinch framework (http://phinch.org) which is the underlying codebase used to generate the "Blackbird" application mentioned in this manuscript.

**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 Dec 2016

**Markus J. Ankenbrand**, University of Würzburg, Würzberg, Germany

Thanks a lot for taking the time to review this article and for the good suggestions for improvement. Find our point by point answers below (original comments in bold):

**Since this project is based on the Phinch framework, I find the "Blackbird" rebranding of the fork to be very problematic. The "Blackbird" instance is really just an updated release of the Phinch framework, with some bug fixes, added features, and implementation of the new BIOM conversion server. The rebranding/renaming is confusing for the end user (see comment by other peer reviewer below), and mistakenly implies a number of scenarios that are not accurate:**
**1) that the authors were involved in the original development of data visualization tools,**
**2) that the Blackbird rebranding and design changes were approved from by the original developers, and**
**3) the "Blackbird" project represents a significant expansion or retooling of the current Phinch framework.**
**I'm fully aware that this is open source software and the authors are free to reuse and share the Phinch codebase, but I don't really see the utility of the "Blackbird" rebranding, and creating an additional web instance that mostly replicates the functionality of http://phinch.org will confuse end users. Since the authors here are really community contributors to the original Phinch project, I would recommend eliminating the "Blackbird" rebranding of the project, and reverting back to Phinch branding (citing the framework release as Phinch v2.0).We will then initiate a pull request to update the bug fixes and integrate the new biojs-io-biom source code to be live on http://phinch.org The visual layout for Phinch (name, logo and visualization layout) was thoughtfully constructed, and the new Blackbird logo and visual modifications will likely interfere with "brand recognition" that should be attributed to the original Phinch framework. Once this pull request is initiated and completed, the "Application" manuscript text should be updated to reflect the live implementation of the conversion library on a v2.0 Phinch framework at phinch.org.**
Thanks for sharing your thoughts on this delicate topic. We are grateful to you for suggesting a more satisfactory solution. As you suggested we prepared the pull request that integrates the additional features into Phinch and removed Blackbird branding from our fork. We look forward to the changes going live on phinch.org. We will use the same procedure for future improvements as long as you are interested in merging them.

**Can you please provide details on how and where the "Blackbird" instance and biom-conversion-server are currently hosted (e.g. Amazon AWS)?**
The biom-conversion-server and the Phinch preview instance are both docker containers currently running on a virtual machine with Ubuntu 16.04 (2GB RAM, 1CPU) on a dedicated server hosted by Hetzner.

**Please list the public landing page for the applications mentioned in the text (in case**

**users want to access these tools directly) - e.g. https://biomcs.iimog.org**
Added links to the manuscript

**The biom-conversion-server does not appear to be backwards compatible (I could not upload and convert a BIOM 1.x file to 2.x format) - this one-way conversion functionality is should be clearly indicated in the first paragraph of the "Application" section. In addition, if users try to upload a BIOM 1.0 file they should be presented with an appropriate error message (I didn't see one - the tool just froze when I attempted to upload a BIOM 1.0 file).**
In general the biom-conversion-server is not limited to one way conversion. Attempts to replicate the described behaviour were not successful so it might be a problem with a specific BIOM file. We are eager to find the cause of this issue and opened a bug report here:
https://github.com/molbiodiv/biom-conversion-server/issues/4
However we need your assistance in tracking down this bug.

**There are other BIOM conversion servers that exist, e.g. implementations within the Galaxy framework - see**
**https://toolshed.g2.bx.psu.edu/repository/display_tool?repository_id=b3ae8ca9317b000e&rer**
**- these alternate tools should be mentioned in the text. How does the biom-conversion-server compare with (and potentially improve on) such Galaxy based tools?**
Thanks for pointing that out. We included the Galaxy biom_convert tool in our discussion.

*Competing Interests:* No competing interests were disclosed.

Reviewer Report 03 October 2016

? **Daniel McDonald**
Department of Pediatrics, University of California, San Diego, La Jolla, CA, USA
**Evan Bolyen**
Center for Microbial Genetics and Genomics, Northern Arizona University

In Ankenbrand *et al*, the authors develop a library to enable interaction with BIOM, a file format common in the microbiome field, from the JavaScript programming language. JavaScript is a staple of web-development, and the ability to interact with BIOM formatted files via JavaScript will facilitate the development of web-based tools for microbiome research. As the authors note, libraries for the interaction BIOM files have only been implemented so far in Python, R and Perl. And while Python and Perl have a strong web presence, they are not natively supported in modern web browsers as JavaScript is, and often

rely on server-side processing as opposed to the client-side paradigms which JavaScript excels at.

**General comments**

- The API provided by BioJS is minimal. Notably, methods for partitioning, collapsing, transforming, filtering and subsampling are not present. While developers will be able to access sample or observation profiles as a whole, the current release of BioJS pushes much of the common manipulation logic onto the consumer of the library.

- The in memory representation of the data following parse by BioJS are either in a dense matrix, or in a dict of keys style sparse representation. As the authors note, specialized methods will need to be created to handle large data efficiently, however the authors may wish to consider placing emphasis instead on specialized data structures such as compressed sparse row or column.

- The highlight with Blackbird is great to see but we were confused by the intention of the Github fork. The codebase suggests that it is more than just a proof of concept to highlight BioJS as there is project-specific branding. Would the authors consider clarifying their position with Blackbird?

- The primary motivator for the development of BIOM-format 2.1.0 were scaling limitations inherent with the JSON-based representation of 1.0.0. Specifically, the "data" key of the JSON string must be parsed in full in order to random access to individual sample or observation data. This removes the possibility of algorithms which depend on efficient random access patterns for data too large for main memory. Additionally, the overhead associated with representing a large JSON object in memory is high. While we acknowledge HDF5 possesses challenges for web-based interaction with these data, it is important to note that the 1.0.0 JSON-based format is not recommended for modern sized studies using hundreds to thousands to tens of thousands of samples.

- The use of the conversion server is very cool and could be taken a step further by layering a light communication API on top to allow a client to request arbitrary samples. This separation would remove the burden of the client needing to read HDF5 formatted files, greatly lower the memory footprint of the client, and likely be more performant than a pure client-side model as the client would only need to know about what it had requested. This expansion of biojs-io-biom, in our opinion, would have the greatest impact for expanding the use of BIOM formatted data within a web application.

**Major**

- When the authors refer to BIOM v2, we believe they are actually referring to BIOM v2.1.0. There are important distinctions between the format versions. Would the authors consider clarifying the minor version number in discussion?

**Minor**

- The two uses of "accession functions" reads awkwardly as these types of methods are generally described as "accessor functions." Would the authors consider revising the phrasing?

**Disclosures**

Daniel McDonald and Evan Bolyen are developers for the BIOM-Format Project.

*Competing Interests:* No competing interests were disclosed.

**We confirm that we have read this submission and 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.**

Author Response 23 Dec 2016

**Markus J. Ankenbrand**, University of Würzburg, Würzberg, Germany

We thank the reviewers for their constructive comments that helped us improve the manuscript. Find our point by point answers below (original comments in bold):

**The API provided by BioJS is minimal. Notably, methods for partitioning, collapsing, transforming, filtering and subsampling are not present. While developers will be able to access sample or observation profiles as a whole, the current release of BioJS pushes much of the common manipulation logic onto the consumer of the library.**
Thanks for pointing that out. We continuously add more functions to make use of our library more convenient. I opened a dedicated issue listing the functions that are present in the python library but lacking in ours (https://github.com/molbiodiv/biojs-io-biom/issues/16). We already implemented functions for transformation, normalization and filtering in order to get more feature complete.

**The in memory representation of the data following parse by BioJS are either in a dense matrix, or in a dict of keys style sparse representation. As the authors note, specialized methods will need to be created to handle large data efficiently, however the authors may wish to consider placing emphasis instead on specialized data structures such as compressed sparse row or column.**
That is a very good point and something we are evaluating at the moment.

**The highlight with Blackbird is great to see but we were confused by the intention of the Github fork. The codebase suggests that it is more than just a proof of concept to highlight BioJS as there is project-specific branding. Would the authors consider clarifying their position with Blackbird?**
After feedback from Holly Bik (Principal Investigator on the Phinch framework) we agreed to remove the Blackbird branding and instead merge our improvements back into Phinch. Therefore, we removed references to Blackbird from the manuscript. For more details see the referee report by Holly Bik (18 Oct 2016) and this discussion on GitHub: https://github.com/PitchInteractiveInc/Phinch/issues/63

**The primary motivator for the development of BIOM-format 2.1.0 were scaling limitations inherent with the JSON-based representation of 1.0.0. Specifically, the "data" key of the JSON string must be parsed in full in order to random access to individual sample or observation data. This removes the possibility of algorithms which depend on efficient random access patterns for data too large for main memory. Additionally, the overhead associated with representing a large JSON object in memory is high. While we acknowledge HDF5 possesses challenges for web-based interaction with these data, it is important to note that the 1.0.0 JSON-based format is not recommended for modern sized studies using hundreds to thousands to tens of thousands of samples.**
This is a valid point. By using the JSON representation for our library we re-introduce the limitations of BIOM-format 1.0. We hope to support the HDF5 format in the future. However even with support of HDF5 loading full tables with tens of thousands of samples into the browser might be too memory intensive. Therefore, the next thing we would like to try is the extension of the conversion server with the communication API as you suggested. We added a short paragraph clearly stating our shortcoming and discussing the possible solution at the end of the Application section.

**The use of the conversion server is very cool and could be taken a step further by layering**

**a light communication API on top to allow a client to request arbitrary samples. This separation would remove the burden of the client needing to read HDF5 formatted files, greatly lower the memory footprint of the client, and likely be more performant than a pure client-side model as the client would only need to know about what it had requested. This expansion of biojs-io-biom, in our opinion, would have the greatest impact for expanding the use of BIOM formatted data within a web application.**

This is a great suggestion and we are eager to work on that for the next major release. We also added this as a future prospect to the manuscript.

**When the authors refer to BIOM v2, we believe they are actually referring to BIOM v2.1.0. There are important distinctions between the format versions. Would the authors consider clarifying the minor version number in discussion?**

We added the minor version number whenever we refer to the BIOM format. We left the patch level out as the documentation on biom-format.org only lists the three versions (1.0, 2.0, 2.1). If you feel that the patch level is relevant as well we will gladly add that, too.

**The two uses of "accession functions" reads awkwardly as these types of methods are generally described as "accessor functions." Would the authors consider revising the phrasing?**

Thanks a lot. We revised the phrasing.

*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

F1000Research