SSI/BioJS–

|  |  |
| --- | --- |
| **Project Title** | SSI/BioJS |
| **Document Title** | SSI/BioJS – |
| **Authorship** | Devasena Inupakutika |
|  |  |
| **Document Filename** |  |
| **Document Version** |  |
| **Document Number** |  |
|  |  |
| **Distribution Classification** |  |
| **Distribution List** | *Project Management Board; Project Team; elsewhere* |
| **Approval List** | *Project Management Board* |

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Personnel** | **Date** | **Comment** | **Version** |
| DI | 14/11/14 | First draft | 0.1 |
|  |  |  |  |

Contents

1 Introduction 4

2 Web presence 4

3 Separate documentation for users and developers based on technical abilities (Novice, Intermediate and Experienced) 4

4 User Experience Review 5

Resources examined 5

Specifications 5

Key points and Recommendations 5

BioJS resources 8

Presence 8

End-user Documentation/ User Guide 9

5 BioJS Terminology 9

6 Expected BioJS Package Structure 10

7 User-developer Experience Review 10

**8 Developer Experience Review………………………………………………………………………….12**

**9 Project Organization and Governance………………………………………………………………14**

# 1 Introduction

BioJS[[1]](#footnote-1) is a community-based standard and repository of JavaScript functional components for representing biological information on the web, and especially for visualising data. BioJS’s development has been prompted by the growing need of bioinformatics visualisation tools to be easily shared, reused and discovered. BioJS’s new version 2.0 has a modular architecture that makes it easy for users with clear APIs and test harness to find a particular functionality without needing to know how it has been built while components can be extended or created for implementing new functionality. This community of developers currently provides a range of functionality that is open access and freely available.

The developers believe BioJS will have a significant impact on both developers and users of web based bioinformatics applications. The short to medium term impact will be on developers and ultimately their funders, as redundant software development will be reduced, documentation and interface strategies will be reused, and components will be recycled through extension of functionality. They also expect a significant motivational effect through community collaboration.

This report presents a review of BioJS and its resources from the perspective of both users and developers. It covers online resources, user and developer documentation, source code and Gold Standards. It proposes recommendations relating to all these aspects.

# 2 Web presence

BioJS has the following online resources:

* A web site[[2]](#footnote-2). This provides an overview of BioJS, a link to their Public Library on Google drive, a contact e-mail, a roadmap linking to BioJS product board on Trello and GitHub activity stream at the bottom.
* A GitHub page[[3]](#footnote-3) with separate repositories for all the BioJS components, Documentation, Events/ Talks and Registry.
* A Google Plus page[[4]](#footnote-4).
* A Twitter feed[[5]](#footnote-5).

These resources are all anonymously accessible.

# 3 Separate documentation for users and developers based on technical abilities (Novice, Intermediate and Experienced)

# 4 User Experience Review

This section presents a review of the latest release of BioJS, BioJS 2.0, from the perspective of its users. This focuses on ease of download and installation of necessary dependencies in order to create or re-use packages using the infrastructure, guidelines and tools built by BioJS along with any prerequisites.

# Resources examined

* BioJS 2.0 registry – Visibility and ease-of-use
* BioJS website as it contains relevant links to its registry[[6]](#footnote-6), downloading, setting up, tutorials, installing and using various BioJS existing and new packages respectively.
* BioJS Tech Fundamental Slides[[7]](#footnote-7)
* BioJS bio npm or web packages
* BioJS[[8]](#footnote-8) and BioJS Education[[9]](#footnote-9) repositories on GitHub
* Getting started with BioJS 2.0 talk and Google Summer of Code summer 2014 experiences[[10]](#footnote-10) talk by Sebastian

# Specifications

**Platform:**

* MAC OS X Version 10.9.2
* 2.6 GHz Intel Core i5

**Software:**

* BioJS Version 2.0 library
* NodeJS Version 0.10.32
* NPM Version 1.4.28
* Git

# Key points and Recommendations

As a part of this evaluation, all the BioJS resources as mentioned in the *Resources reviewed* section were considered which form part of user’s experience. The observations made during the review from users perspective along with corresponding recommendations following usability guidelines are as below:

* The visibility of the status of BioJS packages gives appropriate overview on its registry with list of already available packages.
* The information about BioJS project structure appears to be in a natural and logical order. All the implementation-specific details are open to users. However, novice users/ JavaScript beginners may find it difficult to understand certain terms like “Slush and Slush Generators” and the reason behind their usage. A link to **slushjs[[11]](#footnote-11)** either in the **Package Basics[[12]](#footnote-12)** tutorial or under **How to create a package[[13]](#footnote-13)** section of BioJS’s GitHub repository README would be beneficial.

**Link the core JS concepts or complex technical terms in the tutorials, guides and README’s wherever necessary.**

* Every menu, button, field, widget and links to another web page was clicked, explored and hovered over to check for various tools, help, documentation and tutorials on using and obtaining information about BioJS framework.

**Fixed few typos and broken links re-directing to different web resources. They are present under BioJS project’s GitHub[[14]](#footnote-14) account in the form of pull requests. Most of them are under *edu* repository.**

* BioJS tutorial series, website and registry make objects, actions, options, GitHub activity, recently updated and popular packages visible clearly to users that reduces the amount of information a user has to remember.

**Most of the information[[15]](#footnote-15) about BioJS events, details of each group in the BioJS organisational structure, monthly meeting minutes etc. is open to users and is present as a Public folder on Google Drive.**

* BioJS wiki[[16]](#footnote-16) has got a section *Common error messages[[17]](#footnote-17)* but needs to be updated with some more error messages that users/ JS newbies/ new incoming developers might face while setting up an environment for using and creating BioJS packages.

**Help users recognize, diagnose and recover from errors. Make common and frequently encountered errors clear, comprehensible, precise and also either suggest or provide links to solutions. For e.g. errors that one can face due to difference in versions of software dependencies. In this case, it is worth mentioning clearly in the wiki or README or documentation, the specific version to be used if any.**

* Coding standards are in accordance with JavaScript coding conventions. However, BioJS package folder structure (project skeleton) and names are not consistent. There is mismatch in names that are referred in the tutorials and corresponding names in the package directory, which may cause confusion. After clear inspection, comparing the documentation to actual package and files, it is unclear how up-to-date this information is.

**BioJS edu update is required. Make sure everything is consistent within the software, with other similar packages and platform conventions. This is in the TODO list as mentioned under *edu redesign[[18]](#footnote-18)* GitHub issue.**

**A layout of an expected file structure for BioJS package as later mentioned in this report or as explained in BioJS 1.0 *Installing the repository in the local machine[[19]](#footnote-19)* tutorial is recommended.**

* BioJS gives good user control and freedom. The website and other web resources provide easy navigation, clearly marked exits, undo and redo.
* BioJS implementation-specific details are open to users. Although all the information appears in a natural and logical order but certain terms and their usage doesn’t exactly speak the user’s language. For e.g. In the README.md of the main repository for BioJS, Section 1.3, the terms *package* and *component* are referring to the same thing which might be confusing.

**It is recommended to have a separate *BioJS Terminology* section in wiki as shown later in the *Terminology* section of this document describing such terms which sound similar in general context but might be different when considered specific to BioJS. BioJS team as a part of GitHub issue is already addressing this[[20]](#footnote-20).**

* Help and documentation is present in the form of tutorial series but needs fine tuning of the content based on the recommendations discussed above and certain BioJS documentation guidelines which are covered in the *Documentation guidelines* section. BioJS team and its core developers have been helping users recognise, diagnose and recover from errors as soon they are being reported via GitHub issues. They have been making it clear, comprehensible, precise and suggesting solutions wherever possible.

**The 101 tutorial series[[21]](#footnote-21) is quite self-explanatory. But 102 tutorial series[[22]](#footnote-22) should provide more concise, accurate, clear, easily searchable task-oriented docs[[23]](#footnote-23) centred on concrete list of steps.**

Add feedback and experiences from and to users to the website front page under a separate tab “Who are BioJS users?”. This may help showcase how BioJS and its services have benefited users in their research.

The summary of main recommendations is as follows:

|  |  |
| --- | --- |
| BioJS resources | Presence |
| How to get started | Yes. It is clear from the web site and tutorial series how to start using and developing BioJS packages. README is present in the *edu* tutorial repository stating the initial set-up details clearly. |
| User guide | No. There is no specific user guide or doc available for BioJS. But there are supporting detailed tutorials but they don’t partition user, user-developer and developer information separately. All the tutorials, docs and README’s are common for everyone. Separate user guide is recommended and versions (software dependencies and npm modules) to which they apply to should be listed. |
| Help and support | Yes. All the bugs and issues to be entered on GitHub. BioJS website mentions the way of approaching for help through email, Google groups, GitHub etc. clearly. Great support! **However, as stated above known bugs and issues should be listed under *Common error messages* wiki.** |
| E-mail lists and forums | There are no separate mailing lists or forums but BioJS Google groups mail id is pretty active which is also easily searchable from the website. They have separate Google groups for different themes (Technical, Steering Committee etc.) in their organisation structure. |
| Public bug/ issue tracker | Bugs and issues are mostly reported on GitHub and the core development group is very prompt in providing suggestions and resolving them. |
| Quality of service | Best and reasonable effort. Prompt replies and actions. |
| Contributions | The contribution[[24]](#footnote-24) policy and process is clearly stated on the website as well as the main repository’s README. |

# End-user Documentation/ User Guide

The users of BioJS 0.1 have mostly just been bioinformaticians from EBI due to its lack of framework. Bio web developers built their own tools instead of using BioJS. Components were difficult to integrate and use.

BioJS 2.0 was developed during Summer 2014 as a library of components that are easy to reuse, maintain and deploy on the web. As the project has undergone massive development and great transformation, the user-base is now expected to grow. It is therefore necessary and important to ensure that the library is fully documented, and that it is put up at different levels of technical understanding.

Although all the scripts, modules and packages that are part of BioJS code repository are thoroughly documented but currently, they are created by parsing README.md of every web npm package and displayed at the bottom of every page. Hence, the versions of documentations are sometimes not kept consistent and in sync. The end-user documentation should provide a clear high-level description of what the BioJS does (consolidating the details from various scattered BioJS online resources as covered in “Resources examined” section) and target audience (bio-scientists/ bio-informaticians etc.). The documentation should also explicitly state the assumed background and technical competency for end-users.

A Quick-start guide on how to use a package under User Guide or End-user documentation could be created from the *example snippets* section[[25]](#footnote-25) from BioJS code repository.

# 5 BioJS Terminology

Some of the terminology is covered on their wiki[[26]](#footnote-26). Few more terms as mentioned below would make it more clear for users and developers of BioJS services:

* **Package/ Component:** Component is a BioJS concept, which isan *npm* package. It behaves as a separate independent unit responsible of itself, so developers using it should be able to instantiate it, use it or destroy it. It is written in JavaScript and follows BioJS recommendations. Parsers [[27]](#footnote-27) and Visualisations[[28]](#footnote-28) are two types of components. It is a piece of code (containing multiple modules etc.) that solves a well-defined problem or situation or need.

**It would be better to use either *package* or *component* instead of using both in order to avoid confusion.**

* **Registry:** An interactive webpage of all BioJS packages. Provides ease-of-access and presents status and details of the BioJS packages. The front end of the registry is present at <http://biojs.io/>, back end, which is the BioJS registry worker, is present at <http://workmen.biojs.net/> and the local version of registry is at <https://github.com/biojs/biojs-sniper>. Few other details about registry are present here[[29]](#footnote-29).
* **Module:** It’s a small piece of code that implements a particular functionality at a time that is usually packaged in a single unit so that it can be easily invoked and deployed. For e.g. watchify, browserify, biojs-sniper etc.

# 6 Expected BioJS Package Structure

It is always useful to let users/ developers about the default file structure for a BioJS package when they initially run below command to bootstrap their BioJS project:

**cd <your-project> && slush biojs**

BioJS technical team has already addressed this[[30]](#footnote-30).

# 7 User-developer Experience Review

This part of the review is focused on development tasks using BioJS services and programming components of BioJS. Some of the observed key points with recommendations are listed below:

* Setting-up of development environment to create a BioJS package using BioJS services is easy. Developers have the flexibility to use any of the relevant tools by just following certain Gold Standards/ conventions[[31]](#footnote-31).
* Source code of existing BioJS packages is easily accessible and present in the distributed GitHub repositories.
* Example code snippets are present in the BioJS tutorials that can be easily compiled, customised and used. There are also links to RequireBin[[32]](#footnote-32) for these code snippets, which can be directly built and visualised on web browser.

**A separate wiki with clear information on third-party tools, software (NPM packages to be imported), their versions, location and the process of setting them up would be useful.**

**For e.g. versions of modules as shown below:**

**gulp@3.8.10 node\_modules/gulp**

**├── interpret@0.3.10**

**├── pretty-hrtime@0.2.2**

**├── deprecated@0.0.1**

**├── archy@1.0.0**

**├── minimist@1.1.0**

**├── tildify@1.0.0 (user-home@1.1.1)**

**├── semver@4.2.0**

**├── v8flags@1.0.8**

**├── chalk@0.5.1 (escape-string-regexp@1.0.2, ansi-styles@1.1.0, supports-color@0.2.0, strip-ansi@0.3.0, has-ansi@0.1.0)**

**├── orchestrator@0.3.7 (stream-consume@0.1.0, sequencify@0.0.7, end-of-stream@0.1.5)**

**├── vinyl-fs@0.3.13 (graceful-fs@3.0.5, strip-bom@1.0.0, vinyl@0.4.6, through2@0.6.3, defaults@1.0.0, glob-stream@3.1.18, glob-watcher@0.0.6)**

**└── liftoff@0.13.6 (extend@1.3.0, flagged-respawn@0.3.1, resolve@1.0.0, findup-sync@0.1.3)**

**gulp-jshint@1.9.1 node\_modules/gulp-jshint**

**├── through2@0.6.3 (xtend@4.0.0, readable-stream@1.0.33)**

**├── minimatch@2.0.1 (brace-expansion@1.1.0)**

**├── rcloader@0.1.4 (rcfinder@0.1.8)**

**├── lodash@3.0.1**

**└── jshint@2.6.0 (strip-json-comments@1.0.2, underscore@1.6.0, exit@0.1.2, minimatch@1.0.0, shelljs@0.3.0, console-browserify@1.1.0, cli@0.6.5, htmlparser2@3.8.2)**

* **Copyright and Licensing:**

**Add copyright and license statements to each BioJS package repository. A LICENSE file is present in some of the BioJS package repositories (the ones created using *slush biojs and not all those present on registry*). As BioJS services are public, all the source code repositories should describe the copyright or licensing of their contents. License gets listed while bootstrapping own project with an option to select between Apache 2, MIT and BSD licenses – a developer may not bother to look at them or get confused. This should be clearly documented so developers can understand the implications on extensions they write.**

However, the main BioJS repository mentions the license[[33]](#footnote-33) as Apache 2.

**If MIT and BSD are no longer valid in this case, they should be removed from the options. Also, in case third party software is being used, it should be acknowledged and its license should not be neglected if both are dissimilar.**

* **Contribution Policy:**

BioJS has got contributions policy that allows user-developers to contribute their package/ any other contribution here[[34]](#footnote-34). The contribution policy and the details on this page are comprehensive and understandable.

**However, this link was not easily accessible. Put this on main web page <http://biojs.net/> under the *Get Involved* tab. Also, mention clearly the owner of contributions.**

**8 Developer Experience Review**

This is a review with recommendations as to how the developer experience of BioJS can be improved. This is based upon a review of BioJS’s online resources, Education tutorials, source code and test process, and experience of setting up a local development/ test environment for working on creating a test BioJS package by importing existing packages. This evaluation is focused on development tasks relating to changing the BioJS packages, extend, improve or fixing them.

* **Developer information and Documentation**

Tutorials/ documentation is held in public Git repositories, under edurepository of *biojs* project.Although there is no separate documentation for users and developers (further novice and experienced developers), all the documentation is present at wikis[[35]](#footnote-35) and tutorials consolidated on BioJS Education website generated from the GitHub repositories as base. Most of the documentation is up-to-date and in synchronization with source code [from key points and recommendations under User Experience Review section].

Current BioJS documentation is hosted alongside its related products, which reduced the overhead on developer and his need to search for information from multiple locations. It can also help to ensure that the code and documentation are always in-synch.

BioJS project code repository documentation should have a separate comprehensive Developer’s Guide that could be extracted from common tutorial series (Basic and Advanced) that already exists. The guide should explain that BioJS welcomes contributions in the form of new bio npm packages; bug reports and fixes, documentation and tests.

Required standards for coding, documentation and testing of bio npm packages or a link to their wikis should be outlined before *Submission guidelines* on the Contributions/ Get Involved page (mentioned in the *Contribution Policy* section).

* **Review and update documentation**

Documentation should be reviewed and revised to make sure it is complete and accurate.

* **Style Guide/ Coding Standards**

The style guide outlines naming conventions for products at all levels, from JS code to variables and examples. BioJS follows the guide[[36]](#footnote-36) created by Felix Geisendorfer[[37]](#footnote-37), which is licensed under the CC BY-SA 3.0 license[[38]](#footnote-38) and contributors are directed to these for reference. Example snippets are also provided.

* **Source code management**

JavaScript source code (for packages with keyword biojs) is held in public distributed GitHub repositories, under *biojs-edu[[39]](#footnote-39)* project. The source code is well commented.

All the talks, conference material, Google docs and other manually created documents are already under revision control. Since many of these resources have links to source code or example snippets, holding them under version control made this dependence more manageable. Code is properly arranged in relevant directories clearly pointing to documentation, tutorials and other reference materials.

* **Testing**

Not all the BioJS packages on registry, modules and scripts are subject to testing. The types and levels of testing are not mentioned in the BioJS education material. For e.g. in the tutorial[[40]](#footnote-40) for building a parser using real data, *test* section doesn’t state clearly the level and type (unit test/ integration etc.) of testing.

There is only one BioJS package[[41]](#footnote-41)[[42]](#footnote-42) with both unit and integration tests with comprehensive documentation.

Provide information on a wiki or as a separate tutorial (as a part of tutorial series) on how to write and run Mocha unit tests (with different test cases) and at least one of the automated integration tests (phantomjs).

Implement tests for tutorial component[[43]](#footnote-43) for visualizing snipspector. This could be a good example to show the levels of testing as this imports another snipspector BioJS parser[[44]](#footnote-44) package.

* **Gold Standards**

Gold standards[[45]](#footnote-45) quite similar to style guide outlines conventions that BioJS encourages its users and developers to follow and that will conform to BioJS standards.

All the BioJS packages on its registry don’t follow the checklist mentioned as part of its gold standards except few. Correlating it with licensing and testing as stated earlier, most of the packages are not subject to testing at all and hence; don’t contain unit tests/ integration tests.

Add below recommendations to the checklist:

1. Release code/ package (i.e. Bio NPM package to be published) should pass the test suite when Travis CI platform is taking care of running the tests and deploying BioJS packages.
2. LICENSE: Add copyright and license statements to each BioJS package repository.

* **Dependencies on third-party software**

Some of the third-party dependencies are mentioned in the tutorials e.g. Mocha test framework, Travis-CI continuous integration and deployment platform.

Gold Standards[[46]](#footnote-46), tutorial series and other BioJS web resources provide the list of development, testing and deployment environments along with the links to their sources and websites.

**9 Project Organisation and Governance**

This section highlights and presents guidelines on how to be a good BioJS developer and how GitHub issue trackers can be used to help manage development effectively.

* **Citing BioJS**

You may provide the citation for BioJS on a separate Citation page on website as below:

*The suggested citation for BioJS 2.0 is:*

*<Designers, developers and contributors>. 2014. BioJS 2.0: Represent biological data on the web*

*We suggest in your mentions, you be explicit about the version of BioJS or migrate from BioJS 0.1 to BioJS 2.0.*

Citation page for the SSI can be found on its website[[47]](#footnote-47) for reference, where you may find ways to cite BioJS 2.0 for different media.

* **Use issue tracker actively**

The GitHub repositories for *biojs* and *biojs-edu* projects have issue trackers and are being actively used by BioJS team. Issue trackers[[48]](#footnote-48) for the repositories in these projects are being constantly updated with on-going development tasks, features to be included for future releases and reported issues with appropriate labels e.g. *Meta, enhancement, documentation, wishlist, question* etc.

However, it would be useful to keep list of common and known issues as a part of user guide or novice developer guidelines.

Gather all the known common problems for JavaScript program development or existing BioJS components. Update the issue trackers and Common error wiki page for BioJS project with the known, resolved problems and their workarounds. Users/ new developers can then consult the issue tracker for more information.

* **Impact and Promotion**

BioJS registry lists BioJS components in an interactive list. Each component profile also describes different statistics which includes number of downloads, watchers, issues reported and contributors along with its version and GitHub source code repository link. Information about the usage statistics can be interpreted from the registry of BioJS components.

You may want to contact these contributors and their groups, ask their experiences of using BioJS and for further promotion amongst their contacts.

* **Structure of organisation**

Since BioJS has undergone major transformation in Summer 2014 and is still incorporating minor changes and planning for the redesign of their training/ material or tutorials, any changes to directory structures, naming conventions or project files could cause problems for existing users. Such changes should be considered carefully and stakeholders (including internal committees of BioJS) warned if such changes are implemented, before they redesign and upgrade.

BioJS is already using GitHub and Trello[[49]](#footnote-49) as their project management tools. It would be beneficial for BioJS leadership members to nominate a person from technical/ developer committee as a maintenance co-ordinator and ensure that more than one member from each committee can access project mailing lists and Google groups. This ensures that updates are getting communicated across committees in the project.

1. <http://www.ebi.ac.uk/Tools/biojs/registry/> [↑](#footnote-ref-1)
2. <http://biojs.net/> [↑](#footnote-ref-2)
3. <https://github.com/biojs> [↑](#footnote-ref-3)
4. <https://plus.google.com/+BiojsNet1/posts> [↑](#footnote-ref-4)
5. <https://twitter.com/BiojsLibrary> [↑](#footnote-ref-5)
6. <http://biojs.io/> [↑](#footnote-ref-6)
7. <http://slides.biojs.net/ebi-tech-fundamentals-1/#/8> [↑](#footnote-ref-7)
8. <https://github.com/biojs/> [↑](#footnote-ref-8)
9. <https://github.com/biojs-edu> [↑](#footnote-ref-9)
10. <http://tinyurl.com/biojsv213> [↑](#footnote-ref-10)
11. <http://slushjs.github.io/#/> [↑](#footnote-ref-11)
12. <http://edu.biojs.net/series/101/30_package_basics.html> [↑](#footnote-ref-12)
13. <https://github.com/biojs/biojs#42-how-to-create-a-package> [↑](#footnote-ref-13)
14. <https://github.com/biojs/> [↑](#footnote-ref-14)
15. [https://drive.google.com/folderview?id=0BwVhnCYdaHnGWm1idWxBdi1mTEE&usp=sharing#](https://drive.google.com/folderview?id=0BwVhnCYdaHnGWm1idWxBdi1mTEE&usp=sharing) [↑](#footnote-ref-15)
16. <https://github.com/biojs//biojs/wiki> [↑](#footnote-ref-16)
17. <https://github.com/biojs//biojs/wiki/Common-error-messages> [↑](#footnote-ref-17)
18. <https://github.com/biojs/edu/issues/20> [↑](#footnote-ref-18)
19. <https://docs.google.com/document/d/1oS4M0EKAMre4IDERV61VVTDXJH9F-HGbHxKwNXqjWuo/edit> [↑](#footnote-ref-19)
20. <https://github.com/biojs/edu/issues/20> [↑](#footnote-ref-20)
21. <http://edu.biojs.net/categories/101_tutorial/index.html> [↑](#footnote-ref-21)
22. <http://edu.biojs.net/categories/102_tutorial/index.html> [↑](#footnote-ref-22)
23. <https://github.com/biojs/biojs/issues/131> [↑](#footnote-ref-23)
24. <https://github.com/biojs/biojs#51-get-involved> [↑](#footnote-ref-24)
25. <https://github.com/biojs/biojs#33-how-to-use-snippetsexamples> [↑](#footnote-ref-25)
26. <https://github.com/biojs/biojs/wiki/BioJS-services> [↑](#footnote-ref-26)
27. <https://github.com/biojs/biojs-meta-parser> [↑](#footnote-ref-27)
28. <https://github.com/biojs/biojs-meta-vis> [↑](#footnote-ref-28)
29. <https://github.com/biojs/biojs/wiki/Gotchas-about-the-BioJS-registry> [↑](#footnote-ref-29)
30. <https://github.com/biojs/biojs/wiki/BioJS-package:-file-structure> [↑](#footnote-ref-30)
31. <http://edu.biojs.net/series/102/70_gold_standard.html> [↑](#footnote-ref-31)
32. <http://requirebin.com/> [↑](#footnote-ref-32)
33. <https://github.com/biojs/biojs#5-license> [↑](#footnote-ref-33)
34. <https://github.com/biojs/biojs/blob/master/CONTRIBUTING.md> [↑](#footnote-ref-34)
35. <https://github.com/biojs/biojs/wiki> [↑](#footnote-ref-35)
36. <https://github.com/biojs/style-guide> [↑](#footnote-ref-36)
37. <http://felixge.de/> [↑](#footnote-ref-37)
38. <http://creativecommons.org/licenses/by-sa/3.0/> [↑](#footnote-ref-38)
39. <https://github.com/biojs-edu> [↑](#footnote-ref-39)
40. <http://edu.biojs.net/series/102/50_real_parser.html> [↑](#footnote-ref-40)
41. <https://github.com/greenify/biojs-vis-msa> [↑](#footnote-ref-41)
42. <http://biojs-msa.org/> [↑](#footnote-ref-42)
43. <https://github.com/biojs-edu/biojs-vis-snipspector> [↑](#footnote-ref-43)
44. <https://github.com/biojs-edu/biojs-io-snipspector> [↑](#footnote-ref-44)
45. <http://edu.biojs.net/series/102/70_gold_standard.html> [↑](#footnote-ref-45)
46. <http://edu.biojs.net/series/102/70_gold_standard.html> [↑](#footnote-ref-46)
47. <http://www.software.ac.uk/about/how-cite-us> [↑](#footnote-ref-47)
48. <https://github.com/biojs/biojs/issues> [↑](#footnote-ref-48)
49. <https://trello.com/b/SdfFvEIi/biojs-product-board> [↑](#footnote-ref-49)