CrossMark
click for updates

SOFTWARE TOOL ARTICLE

REVISED

Cytoscape tools for the web age: D3.js and Cytoscape.js exporters [v2; ref status: indexed, http://f1000r.es/4k7]

Keiichiro Ono, Barry Demchak, Trey Ideker

Department of Medicine, University of California, San Diego, La Jolla, CA 92093-0688, USA

v2 **First published:** 01 Jul 2014, **3**:143 (doi: 10.12688/f1000research.4510.1)
Latest published: 28 Oct 2014, **3**:143 (doi: 10.12688/f1000research.4510.2)

Abstract

In this paper we present new data export modules for Cytoscape 3 that can generate network files for Cytoscape.js and D3.js. Cytoscape.js exporter is implemented as a core feature of Cytoscape 3, and D3.js exporter is available as a Cytoscape 3 app. These modules enable users to seamlessly export network and table data sets generated in Cytoscape to popular JavaScript library readable formats. In addition, we implemented template web applications for browser-based interactive network visualization that can be used as basis for complex data visualization applications for bioinformatics research. Example web applications created with these tools demonstrate how Cytoscape works in modern data visualization workflows built with traditional desktop tools and emerging web-based technologies. This interactivity enables researchers more flexibility than with static images, thereby greatly improving the quality of insights researchers can gain from them.

This article is included in the Cytoscape App Collection

Open Peer Review

Referee Status: ✓ ✓

	Invited Referees 1	 2
REVISED **version 2** published 28 Oct 2014	✓ report	
version 1 published 01 Jul 2014	? report	✓ report

1 **Alexander Lex**, Harvard University USA

2 **Jose Villaveces**, Max Planck Institute of Biochemistry Germany

Discuss this article

Comments (0)

Corresponding author: Keiichiro Ono (kono@ucsd.edu)

How to cite this article: Ono K, Demchak B and Ideker T. **Cytoscape tools for the web age: D3.js and Cytoscape.js exporters [v2; ref status: indexed, http://f1000r.es/4k7]** *F1000Research* 2014, **3**:143 (doi: 10.12688/f1000research.4510.2)

Grant information: This work was funded by the National Resource for Network Biology (NRNB) under award numbers P41RR031228 and GM103504 assigned to Trey Ideker.

Competing interests: No competing interests were disclosed.

First published: 01 Jul 2014, **3**:143 (doi: 10.12688/f1000research.4510.1)
First indexed: 01 Dec 2014, **3**:143 (doi: 10.12688/f1000research.4510.2)

REVISED **Amendments from Version 1**

Based on the reviewer's comments, we have fixed some minor grammatical errors, style and structural problems and we have updated the following sections:

1. New figures for side-by-side comparison of original Cytoscape network visualizations and exported web-based visualizations (Figure 3 and Figure 5). This shows how much details are reproduced in the exported web-based visualizations.
2. Added more descriptions for the data set visualized in Figure 4.
3. As a supplement, we have set up a new wiki page for step-by-step instruction how to reproduce the Figure 5. It is linked from the descriptions for the figure.
4. Built a minor updated version 1.0.2 of D3.js Exporter and deployed to the Cytoscape App Store: http://apps.cytoscape.org/apps/d3jsexporter
5. Fixed problems in the examples. One of the reviewers reported problems in the examples page to show the D3.js exports. We have updated and deployed a new web page:

   - Samples as web page: http://keiono.github.io/d3-exporter-sample/

   - Source code repository for the samples: https://github.com/keiono/d3-exporter-sample

See referee reports

Introduction

Cytoscape was born as a GUI-based, Java desktop application in 2003[1,2]. Today, it is a de-facto standard application for biological network analysis and visualization. Around 2005, Java was one of the dominant programming languages for data visualization applications, and many Java-based feature-rich toolkits were developed[3]. However, since they were designed before the re-discovery of Ajax[4], and JavaScript technology was not mature enough to visualize large scientific data sets at the time, developers could not predict the success of JavaScript and related web technologies today. Cytoscape is still an important platform for biological network data integration and analysis, but for data visualization and sharing, we need a new method to take advantage of modern web technologies. Utilizing HTML5 and other emerging web technologies, the Cytoscape Consortium developed a JavaScript library for network visualization called cytoscape.js (http://cytoscape.github.io/cytoscape.js/), the successor of Cytoscape Web[5], to meet the demand from the Cytoscape user community. Although Cytoscape and Cytoscape.js share some of the core concepts, such as Visual Styles or automatic layouts, they are completely independent software packages and there has been no simple way to use Cytoscape data sets in Cytoscape.js.

Public biological data repositories are still growing rapidly and the demand for visualizing those complex biological data sets is high. Traditionally, analysis and visualization of biological data is done by desktop applications, and in most cases, visualizations created by popular libraries (e.g., matplotlib[6], ggplot2[7]) are static images. It is hard to perform exploratory data analysis only with static images, especially for large data sets, because some of the details are lost due to the limited size of printed papers or computer screens. For networks, dynamic features, such as zoom or pan, are particularly important because size of all but trivial networks requires such dynamic features to perform meaningful analysis. Instead of developing custom visualization toolkits for specific data sets, the scientific data visualization community is heading towards web-based technologies to utilize actively developed visualization toolkits such as mpld3 (http://mpld3.github.io/) or Bokeh (http://bokeh.pydata.org/). D3.js[8] is one of the most popular toolkits for creating custom interactive visualizations. If biologists can use the existing powerful desktop application as data authoring tool *and* can publish their results with these emerging web technologies for sharing, it simplifies their workflow for large and complex biological data analysis and visualization.

To bridge the gap between the desktop version of Cytoscape and other web-based data visualization toolkits, we developed Cytoscape modules to generate web-friendly data formats. The goal is to enable shared visualization of Cytoscape networks via a web platform (e.g., browser or HTML5 rendering widgets for tablets). And our strategy is to enable conversion from Cytoscape data objects to a format friendly to web apps, and to provide a template code for creating an interactive web application. In this paper, we present the implementation of Cytoscape data exporters and template web applications and demonstrate how users can publish their data sets as interactive data visualizations with our new tools.

Implementation

The exporter modules were developed for Cytoscape 3. The Cytoscape.js exporter is part of the Cytoscape 3 core distribution and is available as a standard feature. The D3.js exporter is an app, which means developers can write any type of additional JSON exporters as necessary. Exporter modules generate JavaScript Object Notation (JSON) files that are readable by Cytoscape.js and D3.js. To visualize these JSON files as interactive network diagrams, users have to write some JavaScript and HTML5 code to read, map, and render the network and table data in the files. The general structure of code for basic network visualization is common to most use cases. To simplify this visualization task, we provide template HTML5 projects to render the exported JSON files.

Cytoscape.js exporter

Cytoscape.js is a JavaScript library for interactive network visualization developed by the Cytoscape Consortium. Although Cytoscape and Cytoscape.js share core concepts, they have completely independent code bases written in Java and JavaScript. This means there is no binary-level compatibility between these two software packages. The purpose of this exporter is to provide data-level compatibility between these software packages. There are two core functions in this exporter module, which are Cytoscape.js format JSON file generator from Cytoscape networks and tables, and Visual Style converter that creates Cytoscape.js compatible styles as JSON files.

Network/Table to JSON converter

Conversion from Cytoscape networks and tables to Cytoscape.js JSON is done by a serialization module implemented with Jackson, a Java-based JSON parser library (https://github.com/FasterXML/jackson). The converter takes a Cytoscape network object and associated node, edge, and network tables as inputs and converts them into a single JavaScript object represented as JSON. Most of the basic data types are converted into JSON except for nested networks, custom graphics, and node shapes and edge line types, which are only available in Cytoscape.

Visual Style to CSS

In contrast, converting Cytoscape's Visual Styles is a non-trivial process. A Visual Style in Cytoscape is a collection of visual mapping functions, which is a mapping from data to visual variables[9], and default values. Conversion from default visual property values to Cytoscape.js objects is a simple one-to-one mapping. However, some of the Cytoscape visual property values are not compatible with Cytoscape.js, and they will be filtered before the conversion. For example, in Cytoscape, there are three types of visual mapping functions. They are Passthrough, Continuous, and Discrete. Cytoscape.js has the concept of a visual mapping function in its design and it follows standards of CSS and selectors, which is significantly different from the design of Cytoscape visual mapping functions. The converter translates Cytoscape visual mapping functions into combinations of Cytoscape.js selectors and mappers (Figure 1). This translation absorbs differences in design between the two applications and reproduces Cytoscape Visual Styles as JavaScript objects used for styling in Cytoscape.js.

In Cytoscape 3.1.1, some of the visual properties, such as node custom graphics and border line type, are not supported by this translator. Those properties will be supported in the future version of Cytoscape.

D3.js exporter

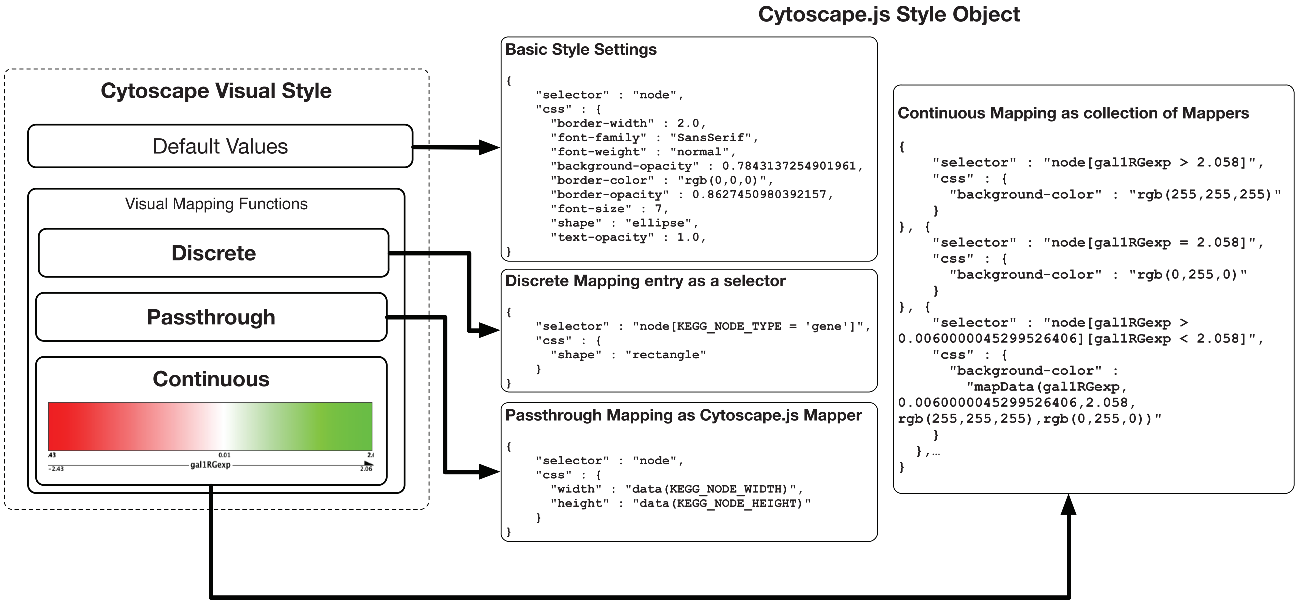
In general, D3.js does not have any specific data format for visualization. Generic CSV/TSV tables can be used for all types of visualizations, and its core provides data loaders for those files. An exception is the graph data format for force-layout, which is the basic preset for visualizing graph data in D3.js (https://github.com/mbostock/d3/wiki/Force-Layout). It uses an ordinal (i.e., zero-based) index of nodes as the unique identifier, and edges are represented as a pair of those indices. The D3.js exporter converts Cytoscape network topology into this force-layout format, and transforms all associated data tables into properties of nodes and edges in the JSON.

A tree data structure is a special kind of graph, and D3.js has various types of preset visualizations for it, such as radial layout, circle packing or Treemap. Cytoscape can visualize trees as node-link diagrams, and if we can export tree data models stored as Cytoscape graph objects into a D3.js compatible format, users can create multiple views for the same data sets using different D3.js visualization presets which could provide new insights for them. To utilize these presets, the exporter generates tree-style JSON for D3.js. The root node of the tree must be specified manually by the user, then the exporter automatically generates tree-style JSON with all associated tables.

Both of the export functions are available under Cytoscape's *File* → *Export* → *Networks*... menu item.

Template projects for visualizing JSON exports

To visualize the exported JSON files, actual web applications are needed to visualize the data. Both Cytoscape.js and D3.js are designed for developers, not for end-users, and developers are expected to write their own custom visualization code. Although they are optimized for custom web-based visualizations, basic components of visualization code, including data loading, mapping, and rendering, are common to most applications. To minimize duplicate efforts to visualize the results from JSON exporters, we developed template web application projects to visualize JSON files generated by Cytoscape. These templates can create basic visualizations of



Basic Style Settings

```
{
    "selector" : "node",
    "css" : {
      "border-width" : 2.0,
      "font-family" : "SansSerif",
      "font-weight" : "normal",
      "background-opacity" : 0.7843137254901961,
      "border-color" : "rgb(0,0,0)",
      "border-opacity" : 0.8627450980392157,
      "font-size" : 7,
      "shape" : "ellipse",
      "text-opacity" : 1.0,
}
```

Discrete Mapping entry as a selector

```
{
    "selector" : "node[KEGG_NODE_TYPE = 'gene']",
    "css" : {
      "shape" : "rectangle"
    }
}
```

Passthrough Mapping as Cytoscape.js Mapper

```
{
    "selector" : "node",
    "css" : {
      "width" : "data(KEGG_NODE_WIDTH)",
      "height" : "data(KEGG_NODE_HEIGHT)"
    }
}
```

Continuous Mapping as collection of Mappers

```
{
    "selector" : "node[gal1RGexp > 2.058]",
    "css" : {
      "background-color" : "rgb(255,255,255)"
    }
}, {
    "selector" : "node[gal1RGexp = 2.058]",
    "css" : {
      "background-color" : "rgb(0,255,0)"
    }
}, {
    "selector" : "node[gal1RGexp >
0.006000000452995264 06][gal1RGexp < 2.058]",
    "css" : {
      "background-color" :
        "mapData(gal1RGexp,
0.0060000004529952640 6,2.058,
rgb(255,255,255),rgb(0,255,0))"
    }
  },…
}
```

Figure 1. Cytoscape Visual Style to Cytoscape.js Style Object conversion.

the JSON files out of the box. To develop these templates, we used standard tools for modern JavaScript development: Node.js (http://nodejs.org/) as runtime for all development tools, Yeoman (http://yeoman.io/) for code scaffolding, and Grunt (http://gruntjs.com) as task runner.

Results

A typical data visualization workflow with our new tools consists of the following four steps. First, users load networks, annotations, and experimental data sets into Cytoscape. Second, utilizing core Cytoscape visualization features, users create custom Visual Styles and layouts. Third, users export all data sets as JSON files, and finally they can create custom web-based, interactive visualizations from the template projects (Figure 2). The original network data visualized in Figure 5 was imported with a Cytoscape app called KEGGScape (http://apps.cytoscape.org/apps/keggscape). The advantage of this workflow is that users can use the large collection of existing Cytoscape apps for data integration and analysis, and the result can be exported as interactive web-based visualizations with the new exporters. Figure 5 shows the TCA Cycle pathway which was generated from a KEGG XML (KGML) file, and its Visual Style was automatically generated from the graphics data in the file. Cytoscape.js exporter can generate web-compatible style and network files directly from the Cytoscape view. Our template code for Cytoscape.js is a simple viewer, and it can be used as a basis for complex data visualization application.

The exporters create JSON files with both network topology and data tables, and users can create complex data visualizations which cannot be achieved with Cytoscape alone. Figure 3 and Figure 4 shows simple network visualizations created with the D3.js force-directed layout (Figure 3) and tree layout (Figure 4). These figures are created with a minimal set of features available in D3.js and they can be used as a "boilerplate" code for custom visualizations. The desktop version of Cytoscape is optimized for rendering node-link, or ball-stick network diagrams, which is only one way to visualize graph data while Cytoscape 3 supports multiple-rendering engines and if developers can implement new rendering engines for new visualizations, such as Treemaps or Chord Diagrams, they can add new visualizations on Cytoscape. This is not a trivial task and as an alternative, our D3.js templates can be used to make prototypes for new visualizations.

Conclusions

In this paper, we presented a new workflow to visualize biological data sets using Cytoscape and modern web-based data visualization libraries. The example visualizations show how users can leverage easy-to-use Cytoscape core features as a part of web-based interactive data visualization publishing workflow.

Future plan

At this point, new features discussed in this paper are designed for developers who can write JavaScript and HTML5 code. End users are also a part of our target audience, and so we will implement the "Export as HTML5 Session" feature as a core Cytoscape feature, which creates a compressed archive file that includes all of the networks, tables, and Visual Styles as JSON along with all JavaScript files to visualize the data as a single-page application.

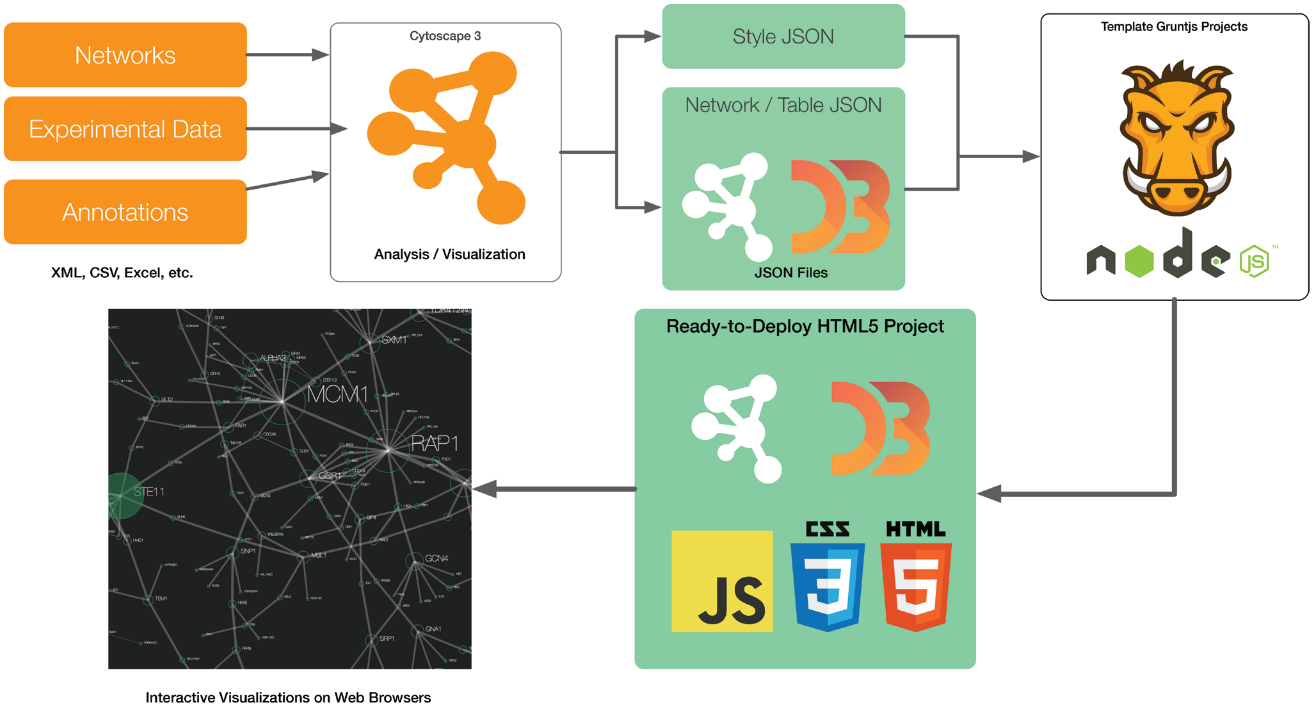


Figure 2. Basic workflow for publishing Cytoscape-generated networks as interactive web visualizations. The combination of Cytoscape 3 and the new JSON exporters can be used as a data integration tool for web-based visualizations. Our sample Grunt project generates a simple template code for visualizing D3.js and Cytoscape.js JSON files.

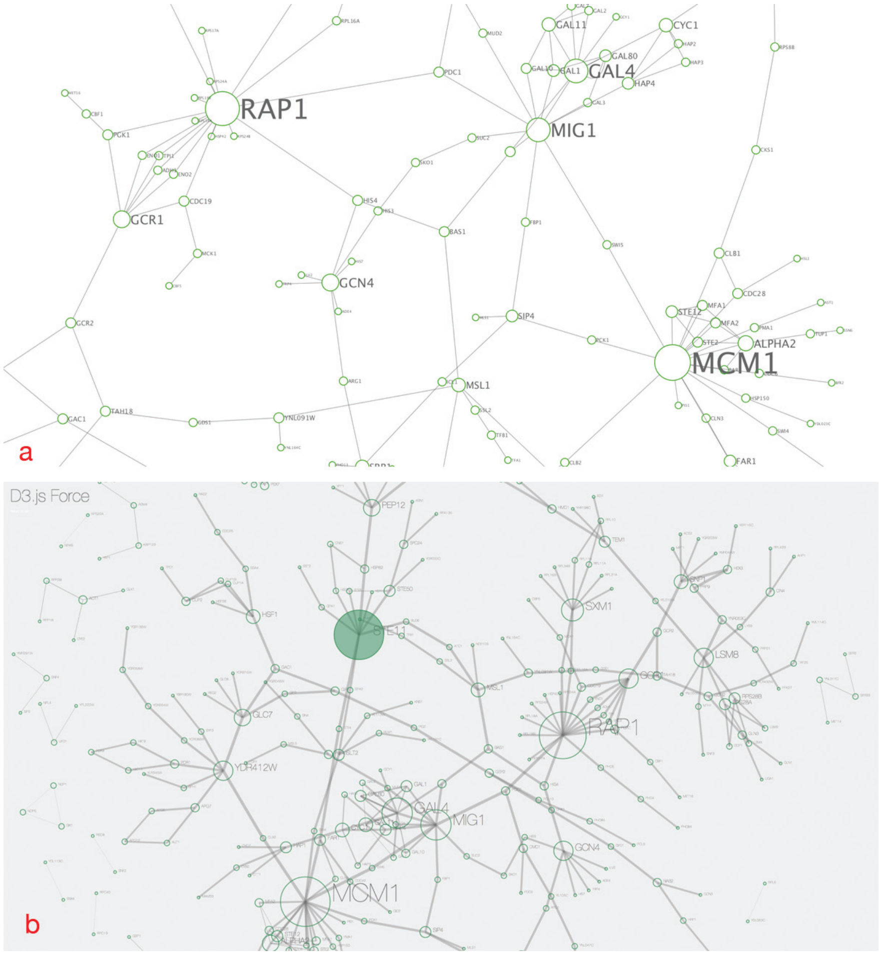


Figure 3. Visualization of a sample yeast protein-protein interaction network (galFiltered.sif from Ideker et al., Science 292:929 2001. Available as a sample file in Cytoscape 3 distribution) by **a**) Cytoscape 3.1.1 and **b**) D3.js force-layout.

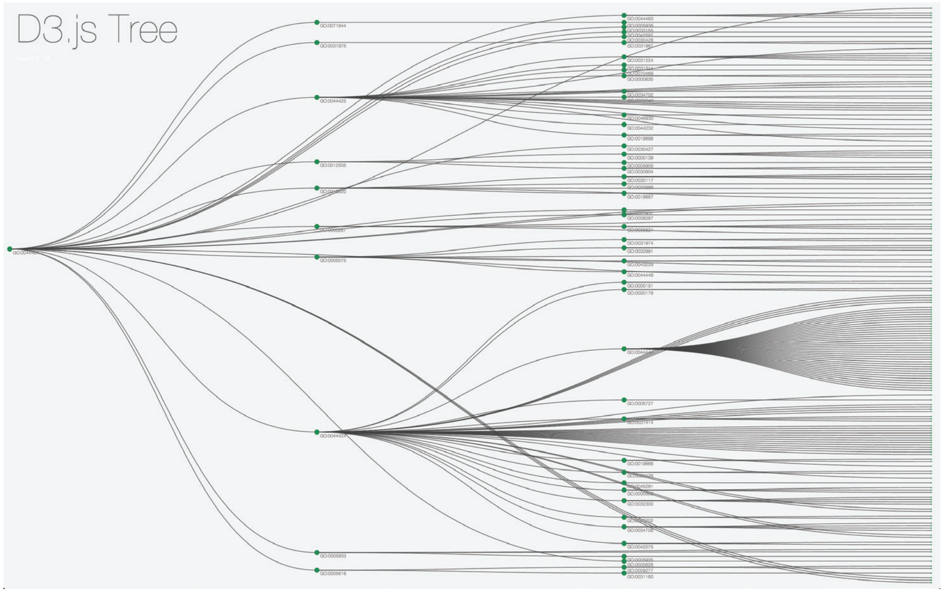


Figure 4. Tree version of Gene Ontology visualized by D3.js tree layout.

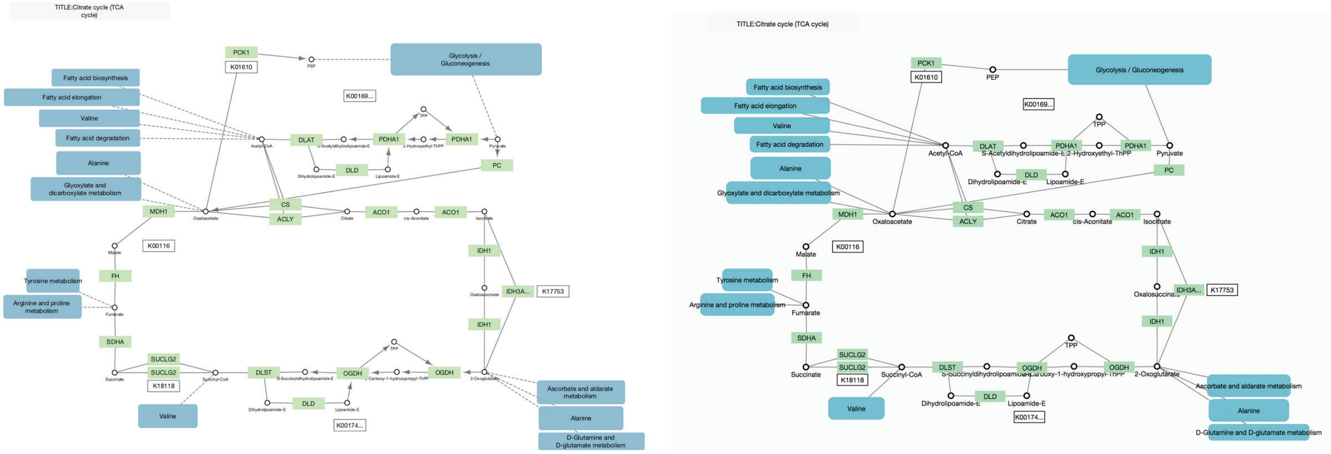


Figure 5. Original network view in Cytoscape 3.1 (left) and same data exported as JSON file and visualized by the template code using Cytoscape.js (right). Original network data was imported to Cytoscape 3 using the KEGGScape app. Cytoscape.js version reproduces almost all details from the original Cytoscape network view. Step-by-step instruction to create web visualization is available at https://github.com/idekerlab/cyjs-exportparent/wiki.

Software availability

Software available from:
http://apps.cytoscape.org/apps/d3jsexporter

Latest source code: https://github.com/keiono/cytoscape-d3

Source code as at the time of publication:
https://github.com/F1000Research/cytoscape-d3/releases/tag/V2.0

Archived source code as at the time of publication:
http://dx.doi.org/10.5281/zenodo.12224[10]

License: MIT License

Template web application code for Cytoscape.js:
https://github.com/idekerlab/cyjs-export-parent

Template web application code for D3.js Exporter:
https://github.com/keiono/d3-exporter-sample

High-resolution images and interactive examples are available at our web sites above.

Author contributions

KO designed and implemented all software packages and wrote this article. BD worked as a project manager and reviewed this article. TI provided funding and supervision.

Competing interests

No competing interests were disclosed.

Grant information

This work was funded by the National Resource for Network Biology (NRNB) under award numbers P41 RR031228 and GM103504 assigned to Trey Ideker.

Acknowledgements

We would like to thank Max Franz and Dr. Gary Bader for their work on the Cytoscape.js JavaScript library.

References

1. Smoot ME, Ono K, Ruscheinski J, *et al.*: **Cytoscape 2.8: new features for data integration and network visualization.** *Bioinformatics.* 2011; **27**(3): 431–432.
PubMed Abstract | Publisher Full Text | Free Full Text

2. Shannon P, Markiel A, Ozier O, *et al.*: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res.* 2003; **13**(11): 2498–2504.
PubMed Abstract | Publisher Full Text | Free Full Text

3. Heer J, Card SK, Landay JA: **prefuse: a toolkit for interactive information visualization**. In *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 421, New York, USA, April 2005. ACM Request Permissions 2005; 421–430.
Publisher Full Text

4. Garrett JJ: **Ajax: A new approach to web applications**. 2005.
Reference Source

5. Lopes CT, Franz M, Kazi F, *et al.*: **Cytoscape Web: an interactive web-based network browser.** *Bioinformatics.* 2010; **26**(18): 2347–2348.
PubMed Abstract | Publisher Full Text | Free Full Text

6. Hunter JD: **Matplotlib: A 2D Graphics Environment.** *Comput Sci Eng.* 2007; **9**(3): 90–95.
Publisher Full Text

7. Wickham H: **ggplot2: elegant graphics for data analysis**. 2009.
Publisher Full Text

8. Bostock M, Ogievetsky V, Heer J: **D³: Data-Driven Documents.** *IEEE Trans Vis Comput Graph.* 2011; **17**(12): 2301–2309.
PubMed Abstract | Publisher Full Text

9. Bertin J: **Semiology of graphics: diagrams, networks, maps**. 1983.
Reference Source

10. Ono K, Demchak B, Ideker T: **F1000Research/cytoscape-d3.** *Zenodo.* 2014.
Data Source

Open Peer Review

Current Referee Status: ☑ ☑

Version 2

Referee Report 01 December 2014

doi:10.5256/f1000research.5911.r6525

☑

Alexander Lex
Visual Computing Group, Harvard University, Cambridge, MA, USA

Dear authors,

Thank you for your revision. My comments on the initial version are completely addressed and I do approve of the current version.

Here are some minor suggestions for improvements of the text:

- To visualize the exported JSON files, actual web applications are needed to visualize the data. -> To visualize the exported JSON files, actual web applications are needed.

- This sentence should probably be split into two:

  The desktop version of Cytoscape is optimized for rendering node-link, or ball-stick network diagrams, which is only one way to visualize graph data while Cytoscape 3 supports multiple-rendering engines and if developers can implement new rendering engines for new visualizations, such as Treemaps or Chord Diagrams, they can add new visualizations on Cytoscape.

- Step-by-step instruction to create web visualization is available at -> Step-by-step instructions to create the web visualization are available at

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 11 September 2014

doi:10.5256/f1000research.4824.r6061

☑

Jose Villaveces

Research Group Computational Biology, Max Planck Institute of Biochemistry, Munich, Germany

The article describes Cytoscape.js and D3.js exporters and a code template generator. Additionally, it provides all the essential information about Cytoscape.js exporter and D3.js exporters and points interested users to the source code for its implementation.

Even though there are a few points that require clarification, the tools are a nice addition to Cytoscape and can be seen as a first step in uniting present day web technologies and the more conventional desktop based visualisation provided by Cytoscape.

Improvements

1. There is no documentation for the template generator, a README.md file with a few examples will be nice.

2. The D3 example listed in the d3jsexporter page is very small (300 x 150) and the 'View Source' button is not working.

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.

Referee Report 01 August 2014

doi:10.5256/f1000research.4824.r5313

Alexander Lex

Visual Computing Group, Harvard University, Cambridge, MA, USA

The article describes an important new development for the Cytoscape project: bridging the gap between complex yet powerful desktop applications to easy-to-use, installation free web application focused on the analysis and presentation. I consider this workflow as an important bridge between the two worlds. The article is technically sound and mostly well written. Some structural comments and concrete suggestions for improving the manuscript are listed below.

What I am currently missing in the manuscript is a description of a workflow for cleaning up the exported visualizations. Especially in the KEGG example, the result is rather messy and clearly requires some additional authoring. How would one go about this process? How long would it take? I would suggest adding another figure that shows a post-cleanup version and a paragraph commenting on the process.

Also, it would be great to see a comparison of the original layout in Cytoscape in addition to the web-based layouts, ideally for all the results figures, so that readers can clearly judge the quality of the approach. This could also put my previous comment into context, as maybe the KEGG layout isn't much better in Cytoscape standalone, since the KEGG annotations typically don't contain sufficient information about the layout.

Structural Suggestions:

- For this argument:

  "However, since they were designed before the re-discovery of Ajax, developers could not predict the success of JavaScript and related web technologies today."

  I would suggest to add that it was simply not feasible at the time since the technology and bandwidth was not available.

- Regarding this sentence:

  "It is hard to perform exploratory data analysis only with static images, especially when visualizing large data sets, because some of the details are lost due to the limited size of printed papers or computer screens."

  I would suggest to also argue that this static approach is especially problematic for networks, as the size of all but trivial networks requires dynamic features to be useful.

- Here:

  "If biologists can use the existing powerful desktop application and these emerging web technologies for data visualizations and sharing, it opens up a new way to understand large and complex biological data sets."

  I think the main point is that it has the potential to combine the best of two worlds: powerful authoring tools and easily shareable interactive results, but I'm not sure whether this is "a new way to understand[ing]". Rather it is a new way of engaging users, through a much simplified process.

- *"via a web platform (e.g., browser)"* -> what would be other web platforms? Or do you mean *"i.e., in the browser"*

- The hierarchy in the Cytoscape.js exporter section isn't clear. The following two captions seem to be of the same level as the root section. I suggest removing the captions completely and simply make the converters bold in the text. Also, these sections sometimes lack clarity. For example, it is not clear what a Cytoscape Table and Network is. I assume that is clear when one knows the software well, but it would be good to clarify this here for readers who are not that familiar with Cytoscape.

- What are *"non-compatible value filterings"*?

- *"The root node of the tree should be specified manually by the user"* -> should or must?

- Caption Figure 4*: "Visualization of a sample network (galFiltered.sif) by D3.js force-layout."* -> What is the data shown here? Should a reader know that from the file name? The file name seems irrelevant here. I would rather add a brief description of the data.

Some grammar and style suggestions:

- *"to popular JavaScript library readable formats"* -> to *"formats readable by popular JavaScript libraries."*.

- *"and Java-based feature-rich toolkits were developed"* -> "*and many Java-based feature-rich toolkits were developed" .*
- *"technologies, Cytoscape Consortium"* -> "*technologies, the Cytoscape Consortium".*
- *"libraries (matplotlib, ggplot2)"* -> "*libraries (e.g., matplotlib, ggplot2)".*
- *"The goal is to enable shared visualization of Cytoscape networks via a web platform (e.g., browser), and our strategy is to enable conversion from Cytoscape data objects to a format friendly to web apps, and to provide a template code for creating an interactive web application."* -> split into multiple sentences.
- *"is part of Cytoscape 3 core distribution"* -> *"is part of the Cytoscape 3 core distribution".*
- *"The D3.js exporter is an app, and this means"* -> "*The D3.js exporter is an app which means"*
- *"The D3.js exporter is an app, and this means"* -> I suggest clarifying "*app*". Is this a cytoscape plugin?
- *"we implemented template HTML5 projects to render the exported JSON files."* -> *"we provide template HTML5 projects to render the exported JSON files.".*
- *"developed by Cytoscape Consortium"* -> *"developed by the Cytoscape Consortium".*
- *"There are two core functions in this exporter module: network/table converter and Visual Style to CSS converter."* -> I would suggest rephrasing this to explain on a high level what these modules do.
- *"except the following: nested networks,"* -> "*except for nested networks,".*
- "*converting Cytoscape Visual Style"* -> is this a noun? or do you mean *"Cytoscape's visual styles".*
- In Figure 1, the code in the *"Continuous Mapping as collection of Mappers"* is unreadable as it is too small.
- *"Once exporters generate JSON files, users need actual web applications to visualize the data."* -> *"To visualize the exported JSON files, actual web applications are needed to visualize the data."*.
- *"visualization code to see the data."* -> *"visualization code."*
- *"Third, all data sets are exported as JSON files, and finally they can create custom web-based, interactive visualizations from the template projects (Figure 2)."* -> This sentence mixes active and passive. I suggest re-writing it in the active voice since the previous sentence is active as well.
- *"The combination of Cytoscape 3 and new JSON exporters can be used as data integration tool for web-based visualizations"* -> *"The combination of Cytoscape 3 and the new JSON exporters can be used as a data integration tool for web-based visualizations.".*

- *"created with D3.js force-directed layout" -> "created with the D3.js force-directed layout".*
- *"such as Treemap or Chord Diagram" -> "such as Treemaps or Chord Diagrams".*
- *"License MIT License" -> "License: MIT License".*
- *"Template code generator website: https://github.com/cytoscape/cyjs-sample" -> "Template code generator website https://github.com/cytoscape/cyjs-sample".*

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 06 Aug 2014

Keiichiro Ono, University of California, San Diego, USA

I completely agree with your point about the missing details for web based KEGG visualization part. I will add more details how to reproduce the visualization with this tool. It will be available as a GitHub Wiki page. The following details will be added to the new version:

- Side-by-side comparison of original Cytoscape visualization and exported version
- Original Cytoscape Session file and exported JSON files as supplemental data
- Link to the instruction wiki page how to reproduce the KEGG visualization

In addition to these changes, I will fix the structural/grammatical issues in the revised version.

Thank you for your helpful suggestions.

Competing Interests: No competing interests were disclosed.

Author Response 25 Nov 2014

Keiichiro Ono, University of California, San Diego, USA

Dear Dr. Lex,

This version 2 includes all updates for your comments for the original version. Also, recently we released new version of Cytoscape (3.2.0), and it should be compatible with this protocol. If you have problems to test this workflow in the latest environment, please let me know.

Best,
Keiichiro Ono

Competing Interests: No competing interests were disclosed.