F1000Research

Check for updates

SOFTWARE TOOL ARTICLE

KnetMaps: a BioJS component to visualize biological knowledge networks [version 1; peer review: 3 approved, 1 approved with reservations]

Ajit Singh [iD], Christopher J. Rawlings [iD], Keywan Hassani-Pak [iD]

Computational and Analytical Sciences, Rothamsted Research, Harpenden, AL5 2JQ, UK

Abstract
KnetMaps is a BioJS component for the interactive visualization of biological knowledge networks. It is well suited for applications that need to visualise complementary, connected and content-rich data in a single view in order to help users to traverse pathways linking entities of interest, for example to go from genotype to phenotype. KnetMaps loads data in JSON format, visualizes the structure and content of knowledge networks using lightweight JavaScript libraries, and supports interactive touch gestures. KnetMaps uses effective visualization techniques to prevent information overload and to allow researchers to progressively build their knowledge.

Keywords
knowledge network, knowledge graph, network visualisation, knowledge discovery, biojs

This article is included in the Agriculture, Food and Nutrition gateway.

This article is included in the Bioinformatics gateway.

BIOJS

This article is included in the BioJS collection.

Open Peer Review

Approval Status ✓ ✓ ? ✓

	1	2	3	4
version 1 17 Oct 2018	✓ view	✓ view	? view	✓ view

1. **Ian Dunham**, Open Targets, EMBL-EBI, Cambridge, UK
   Gareth Peat, Open Targets, EMBL-EBI, Cambridge, UK

2. **Chia-Yi Cheng** [iD], New York University, New York, USA

3. **Nicholas J. Provart** [iD], University of Toronto, Toronto, Canada

4. **Ramil Mauleon** [iD], International Rice Research Institute, Manila, Philippines

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

Corresponding author: Keywan Hassani-Pak (keywan.hassani-pak@rothamsted.ac.uk)

Author roles: Singh A: Formal Analysis, Methodology, Software, Visualization, Writing – Original Draft Preparation; **Rawlings CJ**: Funding Acquisition, Project Administration, Writing – Review & Editing; **Hassani-Pak K**: Conceptualization, Funding Acquisition, Investigation, Methodology, Project Administration, Supervision, Validation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

How to cite this article: Singh A, Rawlings CJ and Hassani-Pak K. **KnetMaps: a BioJS component to visualize biological knowledge networks [version 1; peer review: 3 approved, 1 approved with reservations]** F1000Research 2018, **7**:1651 https://doi.org/10.12688/f1000research.16605.1

First published: 17 Oct 2018, **7**:1651 https://doi.org/10.12688/f1000research.16605.1

Introduction

Networks have been widely used to visually represent complex information in many disciplines, ranging from social sciences (Szell *et al.*, 2010) to engineering, physics, biology, computer science, design and manufacturing (Wang & Alexander, 2015). They fulfil the need to present a system, not only as individual entities but as a whole, by capturing the myriad inter-linked components within the system (Pavlopoulos *et al.*, 2011). Networks are represented as graphs comprising a set of nodes connected by edges. Networks can be homogeneous with all the nodes within the network being of the same type, or heterogeneous with nodes and edges of various types (Sun & Han, 2012). Recently, the term knowledge network or graph has been used frequently in research and business, usually in close association with Semantic Web technologies and linked data. Knowledge networks are increasingly used to model diverse knowledge domains by acquiring and integrating information into an ontology and applying a reasoner to derive new knowledge (Ehrlinger & Wöß, 2016).

A challenge when visualizing knowledge networks is to avoid information congestion and overload that could hinder user experience. The potential richness of data captured in the attributes and density of connections makes it a greater challenge to use standard network visualization tools which often focus on simply visualizing the structure of the network itself (Becker *et al.*, 1995). In molecular biology, there is a wealth of available information, and visualizing all of it at once reduces the value of a visualization or makes it even unusable for analytical purposes, and therefore requires the development of special approaches when visualizing such data (Vehlow *et al.*, 2015).

Previously, our group developed a web-based tool, Ondex Web (Taubert *et al.*, 2014), for visualising knowledge networks generated with the Ondex data integration platform (Kohler *et al.*, 2006). It supported the Ondex exchange language (OXL) and was predominantly used to visualise a biological knowledge domain. However, being a Java-applet and using legacy web technologies, it constantly led to compliance concerns on different web browsers, which hindered its reusability. The advance of modern JavaScript-based data visualisation libraries such as cytoscape.js (Franz *et al.*, 2016) and jQuery (Benedetti & Cranley, 2011) has made it possible for us to learn from our experience with Ondex Web and to develop a new lightweight and reusable component optimised for the visualisation of content-rich knowledge networks.

In this paper we describe KnetMaps (Singh & Hassani-Pak, 2018), an interactive BioJS component to visualise integrated knowledge networks. It is well-suited for applications that require scientists to visualise complementary types of evidence in a single interactive view. KnetMaps is an important visualisation component of KnetMiner where it is used for visualising knowledge networks of crop genomes (Hassani-Pak *et al.*, 2016) and supporting scientists to make informed decisions in gene and trait discovery research. It uses a generic design and hence can be readily embedded in other knowledge discovery applications.

Methods

The KnetMaps component has been developed as part of the KnetMiner software suite and follows the standards set by the BioJS registry. KnetMaps employs a variety of network visualization techniques such as interactive controls, information juxtaposition and data filters. Using effective visualization techniques it prevents information overload and allows researchers to progressively explore and reveal the inter-connected entities within the larger knowledge network (Figure 1).

Visualising knowledge networks

Visualisation of knowledge networks needs to consider two key criteria: i) the heterogeneous and interconnected nature of the network and ii) the content-rich attributes of nodes and edges that cannot be easily displayed as part of the actual network.

Nodes in a biological knowledge network represent entities such as genes, proteins, phenotypes, pathways, publications and ontology terms; connected by edges of various types such as "encodes", "published_in" and "ortholog" (Figure 1C). KnetMaps visualises each node type using a customized combination of shape and colour. Edge types are rendered using a combination of distinct size and colour attributes. The position of nodes and the length of edges is calculated using a force-directed layout that enables connectedness, separation and pattern-based clustering of closely inter-linked entities (Dogrusoz *et al.*, 2009). Labels can be added to nodes and/or edges to enable easier understanding of the underlying data.

To view the potentially rich set of key-value attributes on nodes and edges, we have developed the Item Information panel (Figure 1G). It displays all textual (e.g. abstract, title of a publication) and numeric properties (e.g. accessions, scores and weights) of a selected node or edge in table format, including annotations, detailed descriptions, secondary labels and links to external websites and databases about the selected entity. Users can also use the information displayed in this panel to customize the rendered visualization of node and edge labels to their needs.

Intuitive user interaction

On relevant devices, KnetMaps supports basic touch gestures such as tap, tap-hold, tap-drag and pinch and zoom for user interaction. Users can interact with individual nodes and edges in the rendered network by using standard mouse or touch gestures such as click or tap gesture on a specific node or edge to get a summary of its properties or use the mouse wheel or pinch gesture to manipulate the zoom settings on the network.

Users can right-click or tap-hold on a node or edge to activate a radial context menu (Figure 1F) that provides a range of easy-to-use mechanisms for exploring or manipulating the selected entity. Users can click or tap on a node or edge and view further information such as type, description and annotations, summarised in a dialog box. Users can also tap-drag individual nodes or edges to re-align them within the visible network or tap-hold and reposition the rendered network as a whole.
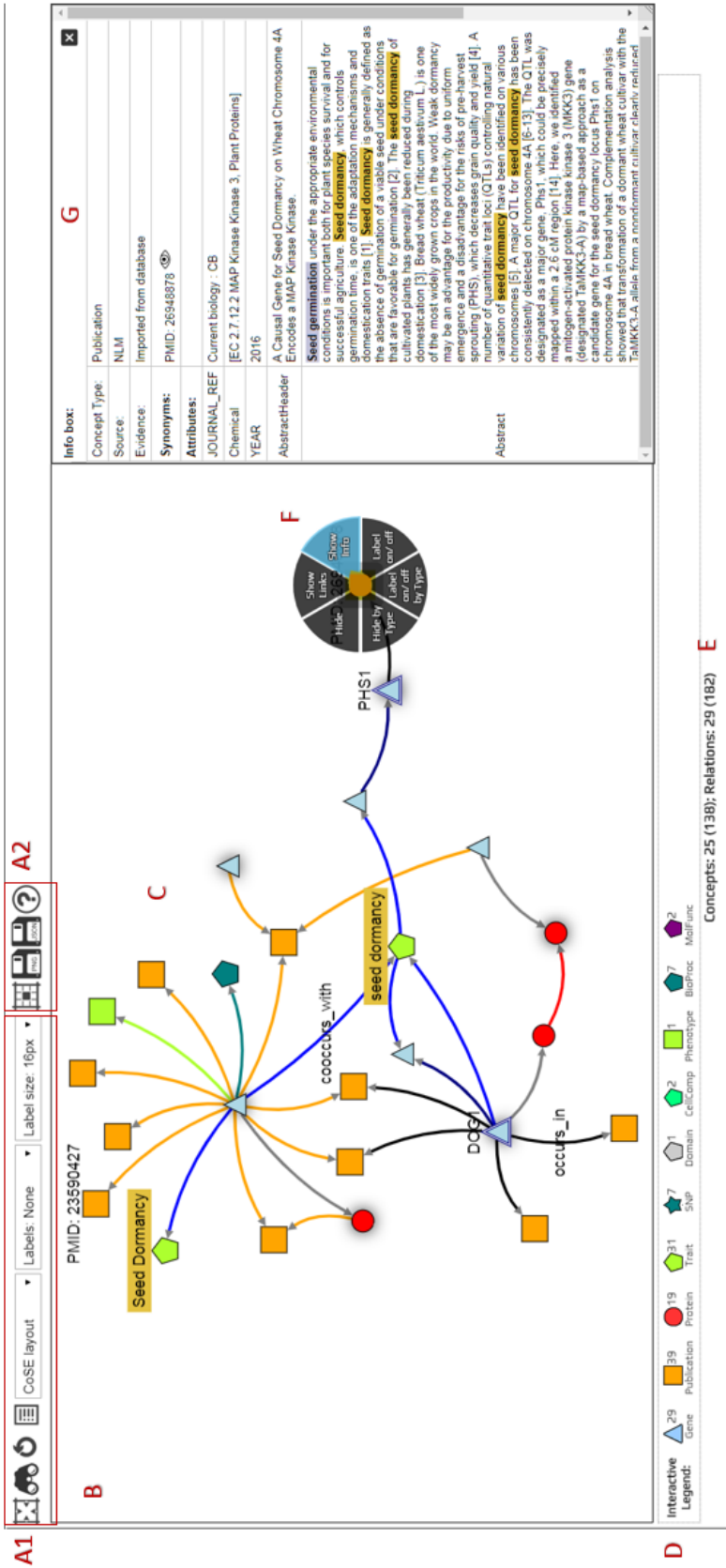
Figure 1. The main components of KnetMaps. (A1) Display configuration (set network layout, node/edge visibility, label visibility and size). **(A2)**. Network export options (cytoscape JSON, PNG). **(B)** Interactive network container (pan and zoom the network), **(C)** Knowledge network rendering (displays nodes and edges of different types). **(D)** Interactive legend (to overlay more linked entities to the visible network). **(E)** Network summary statistics (indicates hidden and total nodes/edges). **(F)** Touch-sensitive context menu (show further information about selected node/edge, hide selected entity or its label, show hidden linked entities). **(G)** Item Information panel (display content-rich attributes of selected node/edge).

The visualized network can be further explored and exported (Figure 1A1) using a variety of menu functions. For example, networks can be exported from KnetMaps as images (in png format) or as cytoscape-compatible JSON which can then be opened in the Cytoscape desktop application (Kohl *et al.*, 2011) for further downstream analysis.

Incremental approach to exploratory analysis

KnetMaps controls information overload in the visible network by providing means to overlay data and extend it in incremental steps, thereby adopting a progressive approach where a subnetwork of interest from the underlying knowledge network is initially visualized and end-users are given the means to add more related information to the visible network. The subnetwork of interest is determined by the application using KnetMaps and passed to it through a "display" attribute in the API/JSON. KnetMaps generates a summary of the number of visible/hidden entities in the knowledge network so that end-users have an overview of what information might be present in the knowledge network but is currently hidden in the visible network. This information is automatically updated each time the user reveals or hides entities from the visible network (Figure 1E).

The first way of adding additional information to the network is by using the *interactive legend* that gives a summary of all node types present in the network, along with a numerical count of the total number of nodes of each type (Figure 1D). For example, clicking on a "Publication" symbol in the legend will add publications linked to visible gene and protein nodes, thereby enabling users to expand the visible knowledge network in real-time.

The second way to add or hide information is by using the *context menu* (Figure 1F). It allows users to hide individual nodes and edges, or hide all nodes and edges of a particular type. This can be useful for removing irrelevant or noisy information from the visible network. Additionally, it allows in- and out-going relations to be added to a selected node when these were initially hidden. This can be useful when a node acts as a knowledge hub, but only a small subset is initially visualized to intentionally prevent information overload, or if a node is part of a larger, more intricate knowledge pathway. In such cases, users can rapidly overlay connected entities within the selected node's neighbourhood onto the visible network to effectively connect the dots and explore the myriad relationships between the network entities.

Implementation

KnetMaps leverages CytoscapeJS v.2.4.7 and jQuery v1.11.2 to visualise knowledge networks. It has been designed in a modular fashion and made available in NPM and BioJS, making it a reusable plug-and-play component within dynamic web applications.

Input data model

KnetMaps loads JSON input data (streamed or locally stored) and renders it as a knowledge network. It uses the cytoscapeJS JSON format specification in which the network is modeled as nested "nodes" and "edges" array objects. Each node or edge entity has a set of required properties such as colour, shape, size, identifier, label, border and visibility. We have extended the cytoscapeJS schema with an additional JSON object to store optional node/edge properties, e.g. abstract and title of a "Publication" node. The separation of required visual properties and optional data specific information, provided a more efficient way of rendering the general network while displaying node/edge specific information on demand, e.g. when the user clicks on a node.

Network rendering

Networks are rendered using a cytoscapeJS-based network stylesheet that maps the set of required JSON properties to the network object. The KnetMaps generator stylesheet sets the shape and colour of a node based on parameters provided for it in the JSON input dataset, e.g., 'shape: data(conceptShape)' and 'background-color: data(conceptColor)' where 'conceptShape' and 'conceptColor' are properties with set values in the input dataset. Developers can customize the stylesheet to replace the supported static cytoscapeJS shapes (such as triangle, round-rectangle, ellipse, pentagon and star) with images. CytoscapeJS selectors have been incorporated in the network stylesheet to filter nodes and edges based on these interactions and add functions that toggle their visual attributes such as highlighting a node or edge when selected and toggling visibility of labels accordingly.

Interactive knowledge display

KnetMaps provides various features for interactive and incremental data exploration by incorporating useful Javascript libraries, such as the cytoscape.js-cxtmenu widget and various force-directed layout libraries to render the knowledge network, including the CoSE layout (Dogrusoz *et al.*, 2009), which is the default network layout used in KnetMaps. Other layouts that can be used by end-users include the physics-based force layout, the CoSE-Bilkent layout that provides additional network topology and geometrical constraints, or static in-built cytoscapeJS layouts, such as the pattern-based circular layout or the concentric layout. KnetMaps packages cytoscapeJS-compatible extensions to these layouts within the application distribution and incorporates optimised settings for each layout within the application itself.

Scalability and performance

Networks of up to 1000 nodes and up to 3000 edges can be visualized in KnetMaps without significant performance degradation or visual delay in layout animations. Visualizing much larger networks (i.e. networks with over 10,000 nodes) increases the initialisation time and can cause jerky or delayed layout animation effects. Some of the rich visual styles used by KnetMaps can be somewhat expensive to render by cytoscapeJS, for example, rendering bezier curved edges.

The KnetMaps code addresses this by providing developers with flexible options to reduce the rendering complexity of the networks. All visual display settings have been made fully customizable to allow developers to tweak element styles such as node

shape, edge curve and node border. Network container settings such as pixel ratio and motion blur can also be similarly easily altered, as can layout parameters such as reducing animation time, decreasing the number of layout iterations to run and disabling animation when rendering very large networks. The default parameters and settings work well in KnetMiner, based on the average sizes of the biological networks (between 300–1000 nodes) that it visualizes. However, customizing these parameters to employ simpler visual settings for larger networks can mitigate performance degradation during rendering.

Operation
KnetMaps has been published to NPM and the BioJS (Gómez *et al.*, 2013) registry, which provides a centralized portal of JavaScript tools and widgets used to analyse and visualize biological data, making it easy for research software users to install KnetMaps and embed within the HTML of their own web pages. The minimum system requirement is a PC with *npm* (part of *Node.js*) installed, a modern web browser with JavaScript enabled and a JSON sample file (see knetmaps/sampleFiles). Once *bower* for managing front-end components and *gulp* for bundling and distribution have been installed to the system (*npm install bower gulp*), following steps are needed to install and bundle KnetMaps:

```
npm install knetmaps
cd knetmaps/
gulp optimise
```

The process will take a few minutes and create the *dist/* folder, containing further *img/*, *js/*, and *css/* subfolders which need to be copied to a web page's root directory. Now, a simple *index.html* can be created to load and visualise the sample JSON dataset:

```
<head>
    <link href="css/knetmaps.css" rel="stylesheet" />
    <script src="js/knetmaps-lib.min.js"></script>
    <script src="js/knetmaps.min.js"></script>
    <script src="sample.json"></script>
    <title>KnetMaps.js demo</title>
</head>

<body>
    <div id="knet-maps"/>
    <script type="text/javascript">KNETMAPS.KnetMaps().
draw('#knet-maps');</script>
</body>
```

Use cases
KnetMaps is used as a network visualization component within tools and platforms that visualize biological knowledge as an interactive network, such as the KnetMiner (Hassani-Pak, 2017) and Daisychain. In KnetMiner, there is a need to visualise query related subsets of a genome-scale knowledge network (Hassani-Pak *et al.*, 2016) in the web-browser. KnetMaps is one of the key components in KnetMiner to visualize and explore integrated information of inter-linked biological entities and processes to help in hypothesis generation and validation, and to accelerate candidate gene discovery. KnetMaps is also part of the KnetMiner Web API and therefore enables collaborators to view knowledge networks for specific genes and keywords from their own applications.

Daisychain is a web application that links genome annotations, aiming to enable researchers working on certain species genes to investigate homologs in other published assemblies via a web interface called Daisychain-Web. The application can be queried using keywords or FASTA sequences with statistical cut-offs and the search results, i.e., links between genes and annotations across similar or identical species or cultivars are visualized as a network using KnetMaps. Daisychain uses KnetMaps out-of-the-box for rendering and visualization and adds further annotation and filtering options to the Item Information panel.

Conclusion
Visualizations are a useful mechanism employed in many disciplines to present information in an intuitive representation that enhances user cognition and helps identify unique patterns and important trends in data. Network formalisms are becoming an increasingly popular means to combine data from inter-connected sources into a concise representation for easier and intuitive exploratory analysis. KnetMaps has been implemented as a fast and lightweight touch-friendly tool for visualizing content-rich, heterogeneous knowledge networks. The implementation uses cytoscapeJS, jQuery and JavaScript extensions for interactive functionality to ensure that low-memory, touch-compatible networks can be rendered in web browsers without the need to write extensive and unwieldy server-side code. Usage of JavaScript ensures rendering compatibility with most web browsers without the need to install any additional software, e.g., Java Applet or Adobe Flash. KnetMaps provides an interactive means to display, filter and overlay networked knowledge, and visually traverse the relationships connecting information within the rendered network. It incorporates a host of visualization techniques such as juxtaposition and superposition to encourage a step-by-step exploration of larger volumes of disparate data, thereby enabling end-users to investigate and analyse inter-linked knowledge in an incremental and intuitive manner.

Data availability
All data underlying the results are available as part of the article and no additional source data are required.

Software availability
Latest source code available from: https://github.com/Rothamsted/knetmaps.js.

Packaged distribution at BioJS: http://biojs.net/#/component/knetmaps.

Packaged distribution at NPM: https://www.npmjs.com/package/knetmaps.

Archived source code as at the time of publication: https://zenodo.org/record/1434144 (Singh & Hassani-Pak, 2018).

Example demonstration of KnetMaps: http://knetminer.rothamsted.ac.uk/KnetMaps/.

License: GNU General Public License v3.0.

References

Becker RA, Eick SG, Wilks AR: **Visualizing Network Data.** *IEEE Transactions on Visualization and Computer Graphics.* 1995; **1**(1): 16–28.
Publisher Full Text

Benedetti R, Cranley R: **Head First JQuery.** O'Reilly Media, Inc. 2011.
Reference Source

Dogrusoz U, Giral E, Cetintas A, *et al.*: **A Layout Algorithm for Undirected Compound Graphs.** *Inf Sci.* 2009; **179**(7): 980–94.
Publisher Full Text

Ehrlinger L, Wöß W: **Towards a Definition of Knowledge Graphs.** *Semantics and Pragmatics.* 2016.
Reference Source

Franz M, Lopes CT, Huck G, *et al.*: **Cytoscape.js: a graph theory library for visualisation and analysis.** *Bioinformatics.* 2016; **32**(2): 309–11.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Gómez J, García LJ, Salazar GA, *et al.*: **BioJS: an open source JavaScript framework for biological data visualization.** *Bioinformatics.* 2013; **29**(8): 1103–4.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Hassani-Pak K: **KnetMiner-An Integrated Data Platform for Gene Mining and Biological Knowledge Discovery.** 2017.
Reference Source

Hassani-Pak K, Castellote M, Esch M, *et al.*: **Developing integrated crop knowledge networks to advance candidate gene discovery.** *Appl Transl Genom.* 2016; **11**: 18–26.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Köhler J, Baumbach J, Taubert J, *et al.*: **Graph-based analysis and visualization of experimental results with ONDEX.** *Bioinformatics.* 2006; **22**(11): 1383–90.
PubMed Abstract | **Publisher Full Text**

Kohl M, Wiese S, Warscheid B: **Cytoscape: software for visualization and analysis of biological networks.** *Methods Mol Biol.* 2011; **696**: 291–303.
PubMed Abstract | **Publisher Full Text**

Pavlopoulos GA, Secrier M, Moschopoulos CN, *et al.*: **Using graph theory to analyze biological networks.** *BioData Min.* 2011; **4**(1): 10.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Singh A, Hassani-Pak K: **Rothamsted/knetmaps.js: v1.0.1 (Version v1.0.1).** *Zenodo.* 2018.
http://www.doi.org/10.5281/zenodo.1434144

Sun Y, Han J: **Mining Heterogeneous Information Networks: Principles and Methodologies.** Morgan & Claypool Publishers. 2012.
Publisher Full Text

Szell M, Lambiotte R, Thurner S: **Multirelational organization of large-scale social networks in an online world.** *Proc Natl Acad Sci U S A.* 2010; **107**(31): 13636–13641.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Taubert J, Hassani-Pak K, Castells-Brooke N, *et al.*: **Ondex Web: web-based visualization and exploration of heterogeneous biological networks.** *Bioinformatics.* 2014; **30**(7): 1034–35.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Vehlow C, Kao DP, Bristow MR, *et al.*: **Visual analysis of biological data-knowledge networks.** *BMC Bioinformatics.* 2015; **16**: 135.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Wang L, Alexander CA: **Big Data in Design and Manufacturing Engineering.** *Am J Eng Appl Sci.* 2015; **8**(2): 223–32.
Publisher Full Text

Open Peer Review

Current Peer Review Status: ✔ ✔ ❓ ✔

Version 1

Reviewer Report 29 January 2019

https://doi.org/10.5256/f1000research.18148.r42475

✔ **Ramil Mauleon** iD

Strategic Innovation Platform, International Rice Research Institute, Manila, Philippines

The functionality provided by KnetMaps is very useful for molecular geneticists (and the bioinformaticians who support them), whose aim is to identify the best candidate genes to use in trait improvement (I am coming from the bioinformatician/geneticist side). Using heterogenous evidences (omics experiment results, literature, function by association inference from sequence identity, GWAS results) is a common exercise in order to get a more informed list of candidate genes to work with. This is the strength of the visualization of Knetmaps: the ability to display the evidences incrementally, and to visualize the connection of evidences into an intuitive network display, which is very helpful for biologists in order to make decisions on which genes to select for further (expensive) experimental validation. Network rendering and response is fast, thus providing an enjoyable end user experience.
I do share one reviewer's feedback on how to revert the network view back to the original state prior to clicking on an icon in the interactive legend, I also seem to miss the step on how to do this.

A suggestion on the implementation section, it would be nice to have a graphical block diagram that shows the steps and inputs required for KnetMaps to be installed in an end-user's system. A statement reporting on installation and functionality of the system under 3 dominant OSs would be appreciated (Linux, Unix/MacOS, MSWindows?), as upfront knowledge for users who wish to install this in their own systems.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, molecular genetics, genomics

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 25 January 2019

https://doi.org/10.5256/f1000research.18148.r42478

Nicholas J. Provart (iD)
Department of Cell and Systems Biology, Centre for the Analysis of Genome Evolution and Function, University of Toronto, Toronto, Canada

In general the concept of the KnetMaps BioJS component is very welcome. I like the functionality that the authors have developed (at least as described in the paper), but I think they could take it up a notch to use additional "visualization techniques to prevent information overload", especially taking advantage of Cytoscape.js's ability to group nodes via the compound node feature – this way, different types of nodes could be grouped (which might be useful for larger networks where things might become quite overloaded). Nodes and edges might receive some additional treatment.

The authors write: "Developers can customize the stylesheet to replace the supported static cytoscapeJS shapes (such as triangle, round-rectangle, ellipse, pentagon and star) with images." It would be nice to actually include a set of image-based icons for the different nodes in the KnetMaps release. Given the multiple types of nodes possible, perhaps a node icon that looks like a page with a few lines on it could symbolize publications, a node icon with an stylized alpha helix could represent the protein etc. I'd also explore using different kinds of edges (straight, curved, dashed) instead of a just a colour scheme.

Sample.json did not seem to be in NPM (as described in the code in the article)? I tried getting

KnetMaps to work within Codepen.io and unkpg.com/knetmaps/ but wasn't quite able to. It would be nice to have this set up so readers could try out the package simply within a browser (by making the "pen" public and sharing the link in the F1000 article). It was almost working, except the JSON file (e.g. Arabidopsis flower network) couldn't be loaded because of a MIME type error and even when I included the JSON object in the script it wasn't working (see code here). Another reason for providing a Codepen link is that there is no preview on the BioJS site.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Data visualization, bioinformatics, gene regulatory network analysis, interaction network visualization, cyberinfrastructure

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.

Reviewer Report 24 January 2019

https://doi.org/10.5256/f1000research.18148.r42477

✓ **Chia-Yi Cheng** iD
Center for Genomics and Systems Biology, New York University, New York, NY, USA

This article describes a package KnetMaps that provides a JavaScript-based tool to visualize

content-rich biological data. KnetMaps is suited for a website with heterogeneous biological information by integrating results in a network format. The examples listed on the demo page (http://knetminer.rothamsted.ac.uk/KnetMaps/) allows users to taste the flavor of KnetMaps. The interface is intuitive and straightforward. Below are suggestions to further enhance the clarity and completeness of the user experience:

1. It would be beneficial for both the developers and end users if the author could provide a step-by-step guide to reproduce one example dataset as on the demo page.
2. The 'interactive legend' feature allows users to add an additional layer to the default network. It is not completely clear to me: i) how was the visible/invisible edges/nodes set in the first place? Ii) is it possible to remove the once added information? For example, clicking on a 'Domain' symbol in the legend will add 'Domains' linked to visible nodes. A user may review and find the information not needed and want to hide those edges. I may have missed it but I did not find a way to cherry pick the symbols once the associated edges displayed.
3. The PNG is working on the demo page (http://knetminer.rothamsted.ac.uk/KnetMaps/) but not the use case pages (http://knetminer.rothamsted.ac.uk/Zea_mays/) (http://daisychain.appliedbioinformatics.com.au/). A blank window popped up when I hit the PNG icon.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: bioinformatics software user and tester

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 01 November 2018

Ian Dunham
Open Targets, EMBL-EBI, Cambridge, UK
Gareth Peat
Open Targets, EMBL-EBI, Cambridge, UK

This is a useful article and software package to help apply cytoscape.js to heterogeneous knowledge graphs. KnetMaps.js adds several features surrounding the cytoscape.js graph visualization, such as the legend, information panel and download functionality. Following the setup guide to run the sample application locally was fairly straightforward.

The software is available online to test at both http://knetminer.rothamsted.ac.uk/KnetMaps/ and http://daisychain.appliedbioinformatics.com.au/. Developers can obtain the source through npm and biojs, where the package is named knetmaps.
KnetMaps.js requires that input data be in a specific format and there are examples of this in the setup guide. The format is easy to understand, but combines data and styling information per node or edge. Separation of these concerns would add clarity and conciseness.

A user wishing to display heterogeneous data, i.e. data containing different node and edge types, might find KnetMaps.js could save them development time. Several features that one might want to add on top of cytoscape.js are provided out of the box, such as the interactive legend, node/edge information panel, PNG/JSON export functionality, filtering and a variety of layout algorithms. However, for any substantial deviation from the UI design provided, a developer might be tempted to start with cytoscape.js directly, which has many online examples and is well documented.

Minor revisions:
- Software
1. PNG export. This functionality does not currently include the interactive legend, which would be useful to explain the static image.
2. File extensions. The sample files have misleading file extensions. For example, ara2.json, which has JavaScript content, should really be a .js file.
3. Automated build step. The running of gulp optimize could be done prior to publishing on npm. This would make it even easier for an end user to embed the necessary .js and .css files.
4. Descriptions for example datasets. There are several example datasets available at http://knetminer.rothamsted.ac.uk/KnetMaps/. It would be helpful to provide a brief description of where the datasets came from and what they contain.
- Manuscript
1. Use of bower. I did not need to use bower to set up the sample application and would suggest removing the reference in the paper.

Longer term suggestions for improvement:

1. Add some unit tests. This would also double as documentation for developers.

2. Use npm to install third party libraries. Several libraries, such as cytoscape, cytoscape-cose-bilkent are not installed using npm. Upgrading them would be more straightforward if the package manager were used.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

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.