LMU|LA
Loyola Marymount
University

Department of Biology

1 LMU Drive, MS 8888
Los Angeles, CA 90045-2659

Kam D. Dahlquist, Ph.D.
kdahlquist@lmu.edu

Tel 310.338.7697
Fax 310.338.4479
www.lmu.edu

Dr. Shawn Gomez
Academic Editor
PeerJ Computer Science

RE: Manuscript CS-2016:05:10823:1:0:NEW, GRNsight: a web application and service for visualizing models of small- to medium-scale gene regulatory networks

August 13, 2016

Dear Dr. Gomez:

We thank the reviewers for their thorough review of both our software and manuscript. We have made changes to both the code and manuscript according to their suggestions; their constructive criticism has led to a much stronger product.

In particular, we have focused on the addition of import and export functionality for the commonly used network exchange formats SIF and GraphML, fixing the bug where the edge weights were not displaying in certain browsers, and making GRNsight available via the BioJS registry. We have clarified in the documentation that nodes can be manually relocated as well as fully documenting the new import/export features. We have also addressed the concerns about the manuscript itself as shown in the response to reviewers.

We now believe that the manuscript is suitable for publication in *PeerJ Computer Science*.

Sincerely,

Kam D. Dahlquist
Digitally signed by Kam D. Dahlquist
DN: cn=Kam D. Dahlquist, o=Loyola Marymount University, ou=Department of Biology, email=kdahlquist@lmu.edu, c=US
Date: 2016.08.13 17:35:48 -07'00'

Kam D. Dahlquist, Ph.D.
Associate Professor of Biology

On behalf of all authors.

Manuscript CS-2016:05:10823:1:0:NEW

GRNsight: a web application and service for visualizing models of small- to medium-scale gene regulatory networks by Dahlquist et al.

Response to Reviewers

Executive Summary

We first want to reiterate what we said in our cover letter: we thank the reviewers for their thorough review of both our software and manuscript. We have made changes to both the code and manuscript according to their suggestions; their constructive criticism has led to a much stronger product.

In this response, we first want to give a general response to what we perceive as the two reviewers' shared concerns (the Executive Summary), followed by a point-by-point response to their specific individual comments. The shared concerns appear to fall into three categories: 1) questioning the raison d'etre for GRNsight; 2) issues with regard to the code; and 3) issues with regard to the manuscript. We will address each of these in turn.

1. <u>GRNsight's raison d'etre given other network visualization software packages</u>

Reviewer 1 notes that:

> *The authors…argue that their tool is aimed at doing one thing well. I agree that this tool does present a network from GRNmap well. However, given the existence of 47 other tools which already do something similar it must be an exceptionally good tool….Ultimately the tool presented here is useful for interpreting the results of GRNmap, but I would be unlikely to use it in any other situation….Alternatively this tool not be considered stand alone from GRNmap.*

Dr. Corpas notes that:

> *I have found disappointing the level of review of other similar network visualisation tools that exist out there…a case should be made so that it is made apparent how GRNsight provides valuable new functionality that is not redundant….I am not sure that the visualisation of 75 nodes or 150 edges is the kind of magnitude that would be valuable to many potential users.*

According to the *PeerJ Computer Science* Aims and Scope:

> PeerJ Computer Science *evaluates articles based only on an objective determination of scientific and methodological soundness, not on subjective determinations of 'impact' or 'readership' for example.*

While we will address these comments in more detail below with regard to the scientific and methodological soundness of our work, we also feel that these remarks could be interpreted as a subjective determination of "impact" of our project, i.e., it should only be published if we can demonstrate that GRNsight is novel and better than every other existing tool in this domain. In fact, we make no such claim. We only claim that GRNsight fulfills the use case for which it was intended and that it is high quality work. Reviewer 1 calls into question whether GRNsight can stand alone as a work separate from GRNmap (in other words, as a "least publishable unit"). In response, we point out that although the requirements for GRNsight were derived from the GRNmap use case and there is valuable interaction between the two groups, GRNsight is a separate project populated by different students using a different technology and following a separate development cycle. Instead of coding a quick "in-house" and "one-off" solution to visualizing data from GRNmap, we took pains to follow open development best practices and be mindful of the features that would make GRNsight applicable to a broader audience than our group, however small that audience turns out to be. (We further note that while we have published the "proof of concept" mathematics behind GRNmap in Dahlquist et al. (2015), we have not yet submitted a manuscript about the GRNmap *software* project because it is still under development.) In the end, GRNsight represents a good example of the scholarship of application (Boyle 1990, *Scholarship reconsidered: Priorities of the Professoriate*, Carnegie Foundation), and as such, should take its place in the scholarly literature.

2. Issues with regard to the code

The two reviewers noted a small number of bugs and feature requests with regard to the code:

1. Feature request: register GRNsight with the BioJS Repository
2. Feature request: allow import and export of network data in other formats besides the Excel workbook format specified by GRNmap
3. Bug: the display of weight values upon edge mouseover only functions for the Firefox browser in Windows 7
4. Feature request: turn on and off the display of all weight values
5. Bug: the bounding box is not large enough for the force spring algorithm to completely relax
6. Feature request: a hierarchical layout may be a better choice for laying out the small- to medium-scale networks displayed by GRNsight
7. Bug: no indication that nodes are no longer responding to the force spring algorithm

Of these, in the 40-day window allotted to revise our manuscript, we focused on the first three issues. We registered GRNsight with the BioJS Repository. We have now implemented import and export of network data in SIF and GraphML format (essential to address the criticisms above of the applicability of GRNsight to a general audience). We have fixed the bug with regard to the display of weights upon edge mouseover in all supported browsers, given that it was a fundamental feature of our software. While we note the bugs and agree with the feature requests listed in items 4-7, they were beyond the scope of the work we could accomplish in the current timeframe. We note that we have clarified the documentation with regard to (7), but also think that it would be better to give the user some visual cue as well. We have written up issues regarding each of these and plan to begin addressing them when the students return for the new semester in a couple of weeks. However, we do not feel that these issues should delay publication of the GRNsight manuscript. More details as to the reasoning and implementation are given in the point-by-point response below.

3. Issues with regard to the manuscript

The reviewers note three main issues with regard to the manuscript itself:

1. Reviewer 1 requests a more thorough discussion of the biological interpretation of the results with regard to the display of a weighted graph.
2. Dr. Corpas requests a discussion of GRNsight with regard to FAIR Principles.
3. They both request both a broader and more detailed comparison of GRNsight with other tools.

We have both added to our discussion of the biological interpretation of the results with regard to the display of a weighted graph and added a section discussing the level of GRNsight compliance with FAIR Principles. The last point, however, requires a more nuanced response. We feel that it is beyond the scope of this article to give a detailed comparison of GRNsight with every other network visualization tool out there. An exhaustive review of all the tools in this domain could be an article in its own right.

We also feel that it is artificial and disingenuous to criticize other tools for not exactly satisfying our requirements when in and of themselves they are good tools. We are fully aware that GRNsight is "yet another tool" for network visualization. But we ask the question, what is wrong with having a diversity of tools that do related things? The first and corresponding author was the project manager for GenMAPP (Dahlquist et al. 2002) and MAPPFinder (Doniger et al. 2003) which were among the first tools to perform pathway analysis of microarray data and Gene Ontology term enrichment analysis, respectively. Now there are hundreds of tools/databases that perform pathway or Gene Ontology analysis in different (however slightly) flavors, including three offshoots from GenMAPP itself. We believe this enriches the scientific community. Upon reflection, we realized that the three main messages that we wanted to deliver with our paper was to 1) of course, describe our product (the software), 2) describe the biological interpretation it enables, and 3) to describe our process. So, in this spirit, we have revamped our introduction and methods sections with a view to highlighting the reasons behind our design choices vis a vis other software like D3 or Cytoscape. We have responded to the specific software comparisons requested by the reviewers below, but, for the most part, have not included the details in the manuscript itself. We hope that these changes have made the rationale for GRNsight clearer in the manuscript. Finally, we note that the visualization software with which we can compare GRNsight at present in 2016 has also evolved from the state it was in circa January 2014 when the GRNsight project began.

Point-by-point response to reviewers

Reviewer 1 (Anonymous)

Experimental design

Dahlquist et al present GRNsight, a service for visualising gene regulatory networks of the 'small-to-medium-scale'. The tool is available via a web application and the source code is also provided. On receiving a manuscript such as this I immediately try and use the application and I compare it to my current preferred tools in the domain. The web application loads quickly and is reasonably intuitive to use. It is not so clear that the force algorithm is disabled on specific nodes once that node has been manually relocated to a new position. This may be described under the help section, but I could not find it although it is mentioned in the manuscript (line 217)

- The disabling of the force algorithm for nodes that are manually moved was previously described on the Documentation page, but we can understand how it was missed because it was near the bottom of the long section on "How GRNsight Displays the Graph". To make this information easier to find, we have created additional subsections on the Documentation page, including one entitled "3c. Manually manipulating the graph" so that the user can more quickly navigate to the information he or she wishes to find.
- For the longer term, based on this feedback, we think it would be helpful to the user to receive some sort of visual cue that the force algorithm is disabled. While that feature is beyond the scope of what we could accomplish during the window for revising our manuscript, we have written it up as an issue for future work (Issue #294, https://github.com/dondi/GRNsight/issues/294).

However,, the only platform for which mouse over on links revealed the underlying weights was firefox on windows 7. Neither safari, chrome not firefox on OSX or Chrome on windows 7 showed the weights

- In our tests, this bug is intermittent, i.e., sometimes the weights show up upon mouseover in Chrome and sometimes they do not. We think that this bug is occurring because we were using the Scalable Vector Graphics (SVG) *title* element, which, when we initiated development, used to default to display as a tooltip. But now that browsers have diverged and new tools like Bootstrap are available, this default behavior is seen less and less. We have addressed this issue by intentionally and programmatically building this feature in with new code and have tested it to work with all supported browsers.

I would like to see an option on the tool for showing or hiding all weights.

- This feature request follows naturally from the bug above, and we agree that it would be a useful feature. However, due to the way that we implemented the new display of weights upon edge mouseover feature, implementing this would require a number of checks that we did not have time to complete before submission of the revisions. We have written it up as an issue for future work (Issue #285, https://github.com/dondi/GRNsight/issues/285).

Validity of the findings

My two preferred tools for laying out networks are BioTapestry (http://www.biotapestry.org/) and YeD (http://www.yworks.com/products/yed) . In neither case could I see a simple way to import the information from the excel file specified by the authors as output from the GRNmap package to reproduce the visualisation that they show in GRNsight. The specific issues are weighted networks � neither of these packages provide a direcct method of importing these data from a matrix in excel. However GRNsight itself does not work directly from a matrix � rather it parses that network through JSON. The central constraint for GRNsight is the file format output by GRNmap. I could write a parser for that matrix into a format such as GraphML and then import that in to YeD and have far more tools available to me.

- We have addressed this concern by implementing the ability to export network data to SIF and GraphML formats (and conversely to import them into GRNsight). The implementation is documented on our Documentation page (http://dondi.github.io/GRNsight/documentation.html) and discussed in the manuscript with regard to FAIR Principles (as recommended by Dr. Corpas below).
- However, since BioTapestry and yED were specifically mentioned by Reviewer 1, we want to further discuss them here.
    - BioTapestry is a sophisticated stand-alone Java program for modeling and visualizing gene regulatory networks as a hierarchy of models that take into account different cell types, spatial domains, environmental conditions, and time points. As such, it is meant to do something quite a bit more complex than what GRNmap or GRNsight does. BioTapestry currently stands at version 7.0.0 and was last updated in September 2014. Source code is available on GitHub (https://github.com/BioTapestry) under the LGPL v2.1 license, and was last updated upon the 2014 release date. It is true that upon submission of our manuscript, there was no easy way to import the adjacency matrix format that GRNmap and GRNsight expects into BioTapestry. BioTapestry can import either SIF (Simple Interaction Format) or CSV files (that have a format specific to the BioTapestry data model). With the implementation of exporting to SIF format, network data can be moved from GRNsight to BioTapestry.
        - In our tests, SIF files exported from GRNsight for unweighted networks are read and displayed properly by BioTapestry v7.0.0.
        - For weighted networks, the porting of network data is less straightforward because the SIF format is not intended to encode the numerical weight parameters. We have decided to export the weight parameters as the "relationship type" in SIF format,

which is normally a string such as "interacts with" or "pd" (for protein→DNA), so as not to lose that data upon export. However, when BioTapestry imports a SIF file, it only allows relationship types of "pos", "neg", and "neu", which then control the display of pointed or blunt arrowheads in BioTapestry. Since we decided to export the weight values themselves instead of converting them to "pos" and "neg", a weighted network will only show pointed arrowheads in BioTapestry.

    - It is of note that when BioTapestry ***exports*** a SIF file, it uses different relationship types than are expected upon import of a SIF file. Upon export it encodes the regulatory relationships as "PROMOTES" or "REPRESSES" instead of "pos" and "neg". Thus, BioTapestry does not consistently read the file exported from itself.
    - When a BioTapestry-exported SIF file is imported into GRNsight, it is treated as an unweighted network. Similarly to how we handle SIF exports for weighted networks, we made the decision to read SIF files with numerical values as the relationship type as weighted networks, but to read SIF files with strings as relationship types as unweighted networks.
    - Thus, even with the implementation of a recognized network exchange format, the movement of network data between GRNsight and BioTapestry is less than perfect, due to differences in the way the different projects prepare and read those files. Editing of the SIF file outside either GRNsight or BioTapestry is required to ensure perfect interoperability. We feel that it is beyond the scope of our work to tailor an import/export function specifically for BioTapestry, given that BioTapestry does not appear to be under active development.

- The yED Graph Editor is a free, but commercially-licensed desktop application for creating network diagrams and is not restricted to the biology domain. It is under active development; during our testing with version 3.16, a new 3.16.1 version was released. yED has many different automatic layout algorithms from which to choose and sophisticated tools for manipulating the visual display of a graph. Indeed, some of the yED layout algorithms are even embedded in Cytoscape v3.4.0. yED supports the import of network data in GraphML and Microsoft Excel format, but not SIF. yED can export network data to GraphML (among other formats). With the implementation of GraphML export, there are now two ways to pass data from GRNsight to yED. yED data can also now be imported into GRNsight via GraphML format.
    - We tried importing the GRNsight demo files "21-genes_31-edges_Schade-data_input.xlsx" and "21-genes_31-edges_Schade-data_estimation_output.xlsx" into yED version 3.16 using the Excel import wizard. They both successfully imported, except that the orientation of the regulators and targets that yED expects for the adjacency matrix is transposed from what GRNsight expects. When we transposed the matrix in Excel, it imported correctly. This is not seamless, but is a fairly trivial operation for experienced Excel users. Note that we originally chose the orientation of the adjacency matrix to match the supplementary data from Lee et al. (2002), from which we derived the network described in Dahlquist et al. (2015).
    - We also implemented the export of GRNsight networks into GraphML format, which can also be read by yED. yED-exported GraphML files can also now be read by GRNsight. Although GraphML is a standard exchange format for network data, its schema allows for some flexibility in implementation. For example, a GraphML file exported from Cytoscape v3.4.0 is not identical to a GraphML file exported from yED v3.16 for the same network, and neither of those is identical to the

implementation we settled upon for GRNsight, although we have attempted to maximize compatibility between GRNsight and both yED and Cytoscape. With regards to yED, both weighted and unweighted GRNsight-exported GraphML networks can be read by yED, except that yED does not display the node labels. This is also an issue when Cytoscape-exported GraphML is imported into yED and has been discussed in a yED online forum. The issue is that yED does not treat the node "id" or "name" elements as node labels, but expects this information to be provided in a *y:NodeLabel* element that is not part of the original GraphML specification. However, upon import, the node label is present as a data field in the Properties View so the user can access the information and manually set the labels if need be. Ironically, the import of node labels is straightforward and successful if the Excel import option is used. Similarly, weight values can be accessed in a data field of the Properties View for weighted networks. Conversely, GRNsight can read an unweighted network exported as GraphML by yED. We wrote specific code to handle the import of the node label from the *y:NodeLabel* element. We note that Cytoscape also has issues reading node labels from yED-exported GraphML.

  - We also encountered a bug in GraphML exported with Cytoscape 3.4.0, which we have reported. The key defining the attribute *weight* is assigned to a *node* element rather than an *edge* element. For the moment, GRNsight does not enforce that *weight* be associated with an *edge* to enable it to read Cytoscape-exported GraphML until the bug can be fixed.
  - One take-home message from this is that even with common formats and data standards, implementations vary, preventing seamless interoperability between tools. GraphML standardizes only the representation of nodes and edges and their directions; all other characteristics, such as names, weights, and other values, are left for others to specify. Although the flexibility is appreciated, flexibility also facilitates divergence, and that is seen here in the discrepancies in how Cytoscape and yED handle GraphML import and export. Thus, we cannot guarantee that GraphML exported from a different program will be read correctly by GRNsight; any issues that do arise will need to be addressed on a case-by-case basis. While we performed test-driven development to add the new import/export functionality (the number of tests reported in the manuscript has gone up), we still need to refactor the server code to repurpose the existing tests of the Excel format for the new SIF and GraphML formats (Issue #300, https://github.com/dondi/GRNsight/issues/300).

The authors themselves recognise this to some degree and argue that their tool is aimed at doing one thing well. I agree that this tool does present a network from GRNmap well. However, given the existence of 47 other tools which already do something similar it must be an exceptionally good tool. I am familiar with cytoscape, but not Gephi and so tried Gephi to visualise a network similar to that presented here. It is possible, but as the authors state non-trivial.

- We responded to this general criticism in our Executive Summary above. We have clarified our rationale for GRNsight vis a vis other tools in the Introduction and Methods.

Whilst I agree that simplicity is key (cognitive load � line 103) II am less clear about the 'understanding the biological results of the model' enabled by GRNsight. The authors discuss these on lines 281-289 giving generalised interpretations of issues such as feed forward motifs, highest in-degree and the regulatory chains. Yet as far as I can see, these are all determined by visual inspection. Furthermore *none of these rely on the weights, which the distinguishing feature of GRNsight. I would be able to identify these more easily in a system such as YeD and do not require the weight information to do so � a simple sif file format givving the directionality between two nodes is sufficient.*

- Yes, you are correct that the insights described in the discussion are derived from visual inspection, which we think is valuable. Such inspection has long been recognized by experts such as Tufte (1983) and Card, Mackinlay, and Shneiderman (1999) as distinct from other forms of purely computational or algorithmic data analysis, and it is this potential that can be derived *specifically* by visual inspection that is targeted by GRNsight. Card, Mackinlay, and Shneiderman (1999) have identified six major ways, documented in earlier literature and empirical studies, by which information visualization amplifies cognition. Tufte's seminal book *The Visual Display of Quantitative Information* (1983) perhaps states and demonstrates it best: "Graphics *reveal* data. Indeed graphics can be more precise and revealing than conventional statistical computations".
- We have expanded the discussion to include interpretation of the weighted results, which, as you point out, is the distinguishing feature of GRNsight.

The visualisation of the weight parameters described in lines 290-322 is the key behaviour here. Figure 5 D,E show clearly the impact of the addition of this weight information, yet E is the clearest visualisation and the only node within it that is located in anything close to its original position is Ace2 � almost every other node has been moved by hand. This revveals a flaw in the implementation of the force-spring layout algorithm as applied in GRNsight. Given the small area of the view port and the constraint that all nodes remain within it, the layout is sub optimal as the force-spring cannot reach its most relaxed state.

- Thank you for pointing this out. We had limited the bounding box for the layout based on what could fit on a typical monitor. While fixing this issue was not within the scope of the work we could accomplish within the window for revising the GRNsight manuscript, we plan to address this issue soon (Issue #159 on GitHub: https://github.com/dondi/GRNsight/issues/159). We plan to increase the size of the bounding box to further maximize the real estate on a typical 24" monitor, as well as giving options for a larger bounding box, which would require the user to scroll to see everything, and a small bounding box for smaller laptop screens. We also plan to implement a zoom feature, which would allow small nodes enough room to relax after which the user could zoom in to view them better.

Furthermore when looking at figure 5 I am always drawn to panels C and F as being the most informative view. Thus I would argue that the force-spring algorithm used here does not provide much benefit to the layout of the network other than separating nodes from one another. The useful layout requires manipulation. Taking the network from E and recreating it in YeD revealed that just a hierarchical clustering and layout gets you closer to E without any manual intervention.

- Thank you for this feedback. While addressing this issue was beyond the scope of what we could accomplish during the revision window, we are looking into using D3 to implement a hierarchical layout for the graph. (Issue #290 on GitHub: https://github.com/dondi/GRNsight/issues/290)

Comments for the Author

Ultimately the tool presented here is useful for interpreting the results of GRNmap, but I would be unlikely to use it in any other situation. As it does not accept a standard input file type, the output of any other network analysis package requires conversion in to the matrix format required here. Similarly, the tool provides no export function (the option in the File menu remained stubbornly greyed out) and so I *can't take a network from GRNsight and utilise it elsewhere. I also can't use GRNsight to convert the GRNmap format to something I might like to use elsewhere. GRNmap itself looks to be a very interesting package and I would like to explore it further, but I would be looking at converting its output into something I could use in a number of other pipelines.*

The authors refer to future features coming in version 2 (lines 323-329). I encourage them to consider implementing at least one standard filetype for displaying graph data within their tool. Be it sif, graphml or even gml, it would significantly increase the utility of the tool as it currently exists. Alternatively this tool not be considered stand alone from GRNmap.

- First, we apologize for the confusion caused by the visibility of the Export menu option, when the functionality had not yet been implemented. We have now made sure that the only menu items to appear are ones that can actually be used.
- Second, as discussed above, we have implemented export of the adjacency matrix into SIF and GraphML formats. We have also implemented import of a network from these formats into GRNsight. We have taken care to maximize compatibility between GRNsight and both yED and Cytoscape. We hope that with these new features, we have increased the utility of GRNsight.

My comments are rather focussed on the tool and its usability, less on the manuscript itself. The manuscript is generally well written and clear. It's weakness lies in the arguments about biological insight derived from the visualisation. If I were to parse the matrix into a graphml file format, I could visualise these networks (complete with weightings and line endings etc) in a wide array of tools and � I would argue �– extract more biological value from the interpretation. I would encourage the authors to expand on this section of the manuscript if possible.

- We have expanded the section on biological insight in the Discussion as suggested.

Reviewer 2 (Manuel Corpas)

Basic reporting

The article "GRNsight: a web application and service for visualizing models of small- to medium-scale gene regulatory networks" is well written and documented.

- Thank you.

I have found disappointing the level of review of other similar network visualisation tools that exist out there. They only provide comparison agains Cytoscape and Gephi. They do not refer (at least in the introduction) to the Cytoscape.js web application, only to the console application. This to me raises concerns in an otherwise exhaustive work, providing ground for not having extensively researched other similar tools in the field, which is quite crowded. A search in the BioJS registry (http://biojs.io/; an NPM based repository of biological web application) of the word "network" retrieves 12 components. Not all of them are necessarily relevant for this publication, but at least some of them should be compared against and a case should be made so that it is made apparent how GRNsight provides valuable new functionality that is not redundant.

- We responded to this general criticism in our Executive Summary above. Here is a rundown of the 12 "network" components currently found in the BioJS registry.
  - cytoscape.js: we will address this one separately below.
  - ols-graphview: This project originated on GitHub on January 10, 2016, and is meant to automatically load ontologies from the backend; it can switch between dynamic and hierarchical layout, collapse nodes, zoom, and move the graph in the bounding box with arrows; edges are labeled with the relationship; it uses the visjs library. This viewer has some nice features that GRNsight might emulate in the future. Aside from the fact that it originated more recently than the origin of the GRNsight project (January 2014), if we had built off of this platform, we would have had to change the way data is imported (user supplied versus database backend) and displayed.
  - complexviewer: This project originated on GitHub on April 28, 2013, and is tailored to display molecular interaction data as undirected graphs. It accepts data in MI-JSON format. Substantive modifications would need to have been made to establish GRNsight functionality on top of this base.
  - giant-api-biojs: This project originated on GitHub on September 13, 2015 and allows visualization of undirected networks. It connects to the GIANT database. This viewer has some nice features that GRNsight might emulate in the future (sliders controlling the edge view). Aside from the fact that it is a more recent than the origin of the GRNsight project, if we had built off of this platform, we would have had to change the way data is imported (user supplied versus database backend) and displayed.
  - mplexviz-ngraph: This project originated on GitHub on September 20, 2015 as a visualisation module for multilayer networks. GRNsight does not deal with multilayer networks.
  - mplexnet: This project also originated on GitHub on September 20, 2015. However, it is a modeling tool related to mplexviz-ngraph above, not a visualization tool.
  - biojs-pcvis: This project originated on GitHub on Dec 14, 2014 and is a web-based binary interaction visualizer that for data in the Pathway Commons database. Aside from the fact that it is a more recent than the origin of the GRNsight project, if we had built off of this platform, we would have had to change the way data is imported (user supplied versus database backend) and displayed.
  - biojs-vis-interactions-heatmap-d3: This project originated on GitHub on February 22, 2015. It draws heatmaps based on D3, which are a modality not required by GRNsight at this time.
  - biojs-vis-interactions-d3: This project originated on GitHub on October 26, 2014 and is meant to visualize protein-protein interactions. It appears to have modified the underlying D3 library to change the shape of the nodes, but has not had any further commits since October 30, 2014. GRNsight has more extensively customized D3 to suit our needs.
  - biojs-vis-hpafeature: This project originated on GitHub on August 10, 2014, but does not display a graph, but annotations from the Human Protein Atlas database.
    - Again, leaving aside the discussion of Cytoscape.js for the moment (which appears as two packages among the 12), we could not have easily adopted any of these packages because 1) they simply do not apply, 2) they are more recent than the origin of our project, 3) they connect to a database back-end instead of user-supplied data, or 4) they would still require modifications to the display style in terms of nodes, directed edges, etc. However, having now reviewed them, some have features we may wish to emulate in terms of zoom, navigation, and sliders.

You do mention Cytoscape.js in line 175 but the justification of creating GRNsight because of the future possibility of implementing other D3.js visualisations is not well founded in my opinion. The http://biojs.io/d/biojs-vis-interactions-d3 network visualisation component does use already D3 for network visualisation.

- As we noted in the Executive Summary, we have revamped our Introduction and Methods to further discuss the implementation decisions we made early in the project. To summarize here:
  - Having defined the requirements for GRNsight, we first looked to see if any existing software satisfied our requirements. At this point we evaluated the Cytoscape stand-alone application, with which the primary author was familiar having been involved in the BioPAX (Demir et al. 2010) and O|B|F communities for a number of years, as well as a few other applications. Cytoscape was already a complex piece of software at that point and we wanted something simpler (due to the cognitive load issues already mentioned in the manuscript). Indeed, one of the undergraduate co-authors who was tasked with evaluating Cytoscape could not get it to run at all (we don't remember what version this was, but it kept crashing her computer whenever she tried to load a graph.) We could have created a Cytoscape plugin/app, but because our primary developers were undergraduates, we felt that it would be too complicated to try to synchronize our work with the larger Cytoscape project. Thus, we settled on creating our own web application. At this point, we again looked around for existing tools. Because the second author was already familiar with D3, we chose that technology. We admit that we did not know of the existence of Cytoscape.js at that point and apologize for the vagueness of the justification of D3 versus Cytoscape.js in the original manuscript. We have removed that sentence and hope with the addition of new text that the rationale is now presented more clearly.
  - As was noted above, the biojs-vis-interactions-d3 project originated on GitHub on October 26, 2014, but has not had any further commits since October 30, 2014. Besides having been originated 10 months after we began the GRNsight project, it appears to be a one-off product that is not under active development. The customizations that it made to D3 only go a small part of the way to what we needed. It was not a viable option for us at the time (or now for that matter).
  - In the Spring 2016 semester, independent of the reviews of this manuscript, we began to explore using Cytoscape.js in order to implement the computation of graph statistics (shortest path between two nodes and betweenness centrality) in GRNsight (Issues #262, 296, 298, 307). This feature is still under development because the out-of-the-box implementation does not yet do what we need it to, which is why we did not discuss it in the manuscript. Nevertheless, we feel that the current feature set of GRNsight stands alone as a publishable unit.

The supplementary figures are very helpful to the understanding of the article.

- Thank you.

Validity of the findings

I am not sure that the visualisation of 75 nodes or 150 edges is the kind of magnitude that would be valuable to many potential users.

- This is a matter of opinion. We feel that GRNmap and GRNsight occupy a niche in between very small biochemically-oriented models and very large whole genome-scale models. Our Google Analytics counter allows us to estimate that approximately 148 files have been uploaded to our server independent of our research group and courses at LMU. While this may be a small number, as we increase the Findability of GRNsight, we will discover how many additional potential users are out there.

I would find it more impressive if the capacity to render nodes was in the order of thousands (even though this may be impractical to visualise and some data reductions might be necessary).

- As noted above, this is not our intended use for GRNsight; other programs already satisfy this need. Our use case for GRNsight was to easily visualize GRNs that GRNmap could usefully model (which is in the range of 15-25 genes/nodes), which is well under the maximum GRNsight allows, which already makes for a very crowded layout. It is not clear that a web service would be able to quickly display the layout of large networks; some stress testing that we have done shows that networks larger than the maximum we allow are slow to render. The value of the edge thickness varying with weight would be lost on a much larger network. However, as noted in response to Reviewer 1, we plan to implement an option to enlarge the bounding box and zoom, which would also facilitate visualizing additional nodes and edges although, perhaps, not on the order of thousands. One potential way to accommodate that would be to implement collapsible nodes, but that is beyond the scope of the current work. We have logged an issue to explore this further in future work (Issue #291, https://github.com/dondi/GRNsight/issues/291).

To me the bits that I have found most useful for the tool are:

- The visualisation is pleasing and intuitive

- The documentation is extensive and easy to read

- The demos allow users to quickly grasp the functionality

- The article is clear and the results show the relevance of the functionality

- The emphasis on testing and best practice are well appreciated although not complete, see below

- Networks can be uploaded via an xlsx file

- Thank you.

I would have also liked some more emphasis on the "findable" and "reusable" aspects of open source software principles. I do believe some mention to "FAIR" principles could be useful: Findable, Accessible, Interoperable, Reusable. This has been done with data sharing (http://www.nature.com/articles/sdata201618) but this applies to software.

- Thank you for this suggestion. We have now included discussion of how well GRNsight complies with FAIR principles in the manuscript.

Comments for the Author

Bioinformatics web tools like GRNsight are published in the literature but they are not made accessible via a centralised repository like the BioJS registry. I would thoroughly recommend authors to make accessible their tool via the BioJS registry. The requirements for making a package accessible through the BioJS registry are minimal:

- The source code has be made available via GitHub

- It has to be made available in the Node.js Package Manager (NPM), the package manager for JavaScript

- The "biojs" keyword has to be included in the "package.json" file of NPM

If the authors had searched the BioJS registry for the keyword "network" they would have found components that could have at least compared against. By not checking the BioJS registry and not including GRNsight in it they have missed the opportunity to increase the exposure of their tool and potentially lose valuable engagement with a community of reference who might even keen to contribute to the code. It is thus important to appear that this tool is not coded in isolation.

- Thank you for this suggestion. We have now made our resource available via NPM and the BioJS registry. We had previously registered our tool with Bioinformatics.org, the Bioinformatics Links Directory at bioinformatics.ca, and the Elixir Tools and Data Services Registry. This is now mentioned in the manuscript as part of the discussion of FAIR Principles.
- We would also like to respond to the statement "*Bioinformatics web tools like GRNsight are published in the literature…*" It was not too long ago that it was more difficult to publish bioinformatics software papers and receive credit for open source development work. It was for that reason that the primary author edited a *Proceedings* published in *BMC Bioinformatics* for the Bioinformatics Open Source Conference (BOSC) in 2010, which she chaired. BOSC is one of the main ways she stays connected to recent developments in the open source community; she missed BOSC in 2014 and 2015 when BioJS was presented because of the personal circumstances of being unable to travel due to having a small child at home. However, she did present GRNsight at BOSC/ISMB a month ago. We would welcome outside contributions to the code, but we are not depending on it. As has been discussed many times at BOSC and in other venues (including by yourself), growing and sustaining an open source community is difficult work (as is maintaining a current list of bioinformatics tools). "*It is thus important to appear that this tool is not coded in isolation.*" We do not feel particularly isolated, especially since the GRNsight developers are interacting with the GRNmap "customers" in an "eat your own dogfood" way. With the GitHub repository, mechanisms are also in place to receive feedback and assistance from a wider developer community. With a steady pipeline of students coming into the GRNsight project, extensive documentation, an established process of onboarding new personnel, and mentoring, we plan to maintain GRNsight for years to come.