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**

**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)
  + This information was included on the Documentation page, but we can understand how it was missed because it was near the bottom of the section on “How GRNsight Displays the Graph”. To make this information easier to find, we have created additional subsections on the Documentation page so that the user can more quickly navigate to the information he or she wishes to find.
* 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
  + This bug occurred because we were using the web page attribute “title”, 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. It was missed by the primary author because she primarily uses Firefox on Windows 7 for testing. 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. [Issue #284].
* I would like to see an option on the tool for showing or hiding all weights.
  + This feature request followed naturally from fixing the bug above and has now been implemented. Fixing the bug listed above, we made the design decision to have all weights display upon mouseover, not just the selected weight. The user now has the option to display all weights, only display all the weights upon mouseover, or hide all the weights. The option can be accessed under the Edit > Preferences menu or via radio button in the main user interface and is explained on the Documentation page. [Issue #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.
  + 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. 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). We have implemented the ability to export the GRNsight native Excel format (adjacency matrix) into SIF format that can subsequently be read by BioTapestry. [Issue #287] This feature can be accessed from the menu item File > Export Data > To Unweighted SIF or File > Export Data > To Weighted SIF and is described on the Documentation page.
    - The export is straightforward for unweighted networks.
    - For weighted networks, it is less straightforward because the SIF format is not intended to encode the numerical weight parameters that set the thickness of the edges. 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).
      * However, when BioTapestry imports a SIF file, it only allows relationship types of “pos”, “neg”, and “neu”, which then controls the display of pointed or blunt arrowheads in BioTapestry. There is no way (that we can tell) to use a SIF file to control the thicknesses/colors of the edges in BioTapestry. We decided to export the weight values themselves instead of converting them to “pos” and “neg” in order to not lose information.
      * It is of note that when BioTapestry ***exports*** a SIF file, it uses different 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”. This calls into question the standardization of the SIF format even when used within the same program. When a BioTapestry-exported SIF file is imported into GRNsight, it is treated as an unweighted network. We similarly 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.
  + We tried importing the GRNsight demo file “21-genes\_31-edges\_Schade-data\_input.xlsx” into yED version 3.16 using the Excel import wizard. It 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 I transposed the matrix in Excel, it imported correctly. 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. [More here.]
* 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 are well aware that GRNsight is “yet another” graph layout tool and that as such, creating a new tool instead of using an existing one demands justification. We did not want to bog down our manuscript with a lengthy description of why each of the other tools did not serve our needs because we did not want to unduly criticize other tools for not serving our exact needs when they perform the function for which they were intended well.
* 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 [Dondi help with the Tufte reference here.] We have also 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.
* 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. We are looking into implementing 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 been implemented yet. 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 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 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. 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.

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 response to Reviewer 1, We are well aware that GRNsight is “yet another” graph layout tool and that as such, creating a new tool instead of using an existing one demands justification. We did not want to bog down our manuscript with a lengthy description of why each of the other tools did not serve our needs because we did not want to unduly criticize other tools for not serving our exact needs when they perform the function for which they were intended well. We have now augmented our discussion of what other tools are available and how they compare to GRNsight in both the Introduction and Discussion sections. We did have a very specific use case in mind when establishing GRNsight. We could have focused our efforts on modifying an existing program, but instead chose to start a new project. Part of our consideration in doing so was that fact that we run a research group composed entirely of undergraduates and that GRNsight was initiated as Britain Southwick’s senior capstone project for his computer science major. We believe that he would have had less success in completing a project within the space of a semester if he were to try to plug into an existing open source project; indeed, one of the requirements of the capstone experience is establishing a new project from the ground up [@dondi, is this correct?]. In our group we also maintain close communication between the GRNsight coding team and the GRNmap data analysis team (who acts as GRNsight “customers”). This is facilitated by local face-to-face interaction between students on a time-scale that can be maintained during an otherwise busy semester.
* 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. 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).
  + 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 would already make for a very crowded layout. 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.
* 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 GRNsight complies with FAIR principles.

**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.