WEB TOOL

HeatMapViewer: interactive display of 2D data in biology
[version 1; referees: 2 approved]

Guy Yachdav\textsuperscript{1-3}, Maximilian Hecht\textsuperscript{1,2}, Metsada Pasmanik-Chor\textsuperscript{4}, Adva Yeheskel\textsuperscript{4}, Burkhard Rost\textsuperscript{1-3}

\textsuperscript{1}TUM, Department of Informatics, Bioinformatics & Computational Biology, 5748 Garching/ Munich, Germany
\textsuperscript{2}TUM Graduate School of Information Science in Health (GSISH), 85748 Garching/Munich, Germany
\textsuperscript{3}Biosof LLC, New York, NY, 10001, USA
\textsuperscript{4}Bioinformatics Unit, G.S.W. Faculty of Life Sciences, Tel Aviv University, Tel Aviv, 69978, Israel

Abstract

Summary: The HeatMapViewer is a BioJS component that lays-out and renders two-dimensional (2D) plots or heat maps that are ideally suited to visualize matrix formatted data in biology such as for the display of microarray experiments or the outcome of mutational studies and the study of SNP-like sequence variants. It can be easily integrated into documents and provides a powerful, interactive way to visualize heat maps in web applications. The software uses a scalable graphics technology that adapts the visualization component to any required resolution, a useful feature for a presentation with many different data-points. The component can be applied to present various biological data types. Here, we present two such cases – showing gene expression data and visualizing mutability landscape analysis.


This article is included in the BioJS collection.
Introduction

Biological data are often organized into matrices in which the rows signify different items of interest (a gene, a subject, a probe or a position in a sequence), while the columns describe different experiments, variations, or samples. Matrices are easy to process by algorithms. In contrast, the details in large matrices are often, at best, challenging for experts who want to “understand” the data. The information in matrices is usually better digested if presented by 3D plots or heat maps. Heat maps are essentially simplified versions of 3D plots that replace the 3rd dimension with color gradients, thereby conveniently displaying the information contained in matrices. Such heat maps allow for easy visual differentiation between high and low values in a matrix.

Such heat maps are, for example, commonly used to display microarray data as they quickly show which genes (rows) are differentially expressed under some conditions (columns). Microarray technologies utilize arrays of probes located on different exons for each gene and can be helpful in determining gene function by measuring transcription and translation levels under certain experimental conditions. The expression values for the differential expression may be presented at the exon level, correlated with protein domains, and may help to decipher a complex gene expression pattern.

Heat maps can also display the effect of point mutations (single amino acid substitutions, or non-synonymous Single Nucleotide Polymorphisms – nsSNPs). Through the application of methods that predict the impact of mutations\(^1\)\(^-\)\(^4\) we can expand from the view of single variants to the level of sketching the entire mutability landscape\(^5\). This mutability landscape is defined by the impact of substituting every residue at each position in a protein by each of the 19 non-native amino acids. The resulting predictions can then be shown in a heat map in order to visualize the impact of each substitution. Regions where mutations have a high average effect (i.e. where almost every substitution is predicted to alter protein function) are especially interesting as these are likely to be of particular and direct importance for protein function.

We developed HeatMapViewer as a BioJS component that can easily be used, reused and, if needed, extended to display matrix data. BioJS\(^6\) is an open source JavaScript library of components for visualization of biological data on the web. As a JavaScript component, the HeatMapViewer is very flexible, interactive and webready. Previous libraries generating graphical HeatMaps render either static images\(^7\) or are highly specialized and cannot be reused\(^8\). To the best of our knowledge, this is the first client-side modular component to visualize matrices that can be integrated into other web applications in a standard manner.

The HeatMapViewer component

HeatMapViewer uses the D3\(^9\) JavaScript library to render Scalable Vector Graphics (SVG) objects. SVG technology is now standardized and native to modern web browsers (e.g. Mozilla, Chrome, Safari). The component accepts a simple JSON object containing the data matrix that will be rendered. A secondary JSON object contains configuration directions such as the target DIV element onto which the component will be rendered, the data range to be shown, the color scheme to be used for the component, the size of the canvas showing the component and the minimum cell size (by default these last two options can be computed automatically).

The HeatMapViewer component automatically renders a heat map based on the input data object and the pre-set color-scheme. Positioning and layout are automatically calculated given the available browser window size. If presenting the entire heat map requires individual cells to be smaller than a given threshold, a secondary panel is automatically rendered to show a zoom-in version of a local segment in the heat map. This zoom-in panel is presented right under the main heat map panel. The labels for the X-axis and Y-axis are laid out above the top row and next to the left column. The component provides a user control in the form of a frame that can be dragged along the main heat map to determine which area of the heat map should appear in the zoom-in panel. Additionally, a scale bar is presented to show the value ranges and which colors correspond to those values. Finally, each cell in the heat map is associated with a mouse-over event that pops-up tooltips showing the data-value of the cell.

The HeatMapViewer component can be obtained from the BioJS registry at https://github.com/biojs/biojs. For users wishing to test the component’s capabilities to generate heat map plots for their data without downloading and installing the component, we have set up a server: http://www.rostlab.org/services/heatmap-viewer. The server allows users to upload their data in Comma Separated Values (CSV) format and then renders a heat map on the screen. The server also allows exporting the resulting graphics rendering it into an image.

Application use-cases and examples

Eye disease Retinitis Pigmentosa (RP)

The rhodopsin gene encodes a protein of the outer photoreceptor segment that is essential for the visual transduction cascade. Since 1989, many mutations in the rhodopsin gene have been found to be involved in the eye disease Retinitis Pigmentosa (also known as Retinopathy pigmentosa or simply RP\(^10\)(1). RP is a hereditary disease causing retinal degeneration and thereby destroying photoreceptors; this results in severe vision impairment or even blindness.

A typical study of such a hereditary disease might begin with a protocol as follows. According to the UCSC genome browser\(^11\), human rhodopsin (RHO, RefSeq: NM_000539.3) consists of 5 exons (located on chr3:129,247,482-129,254,187). The total gene length is 6706 bps (base pairs/nucleotides). The coding region (chr3:129,247,577-129,252,561; i.e. extending over 4985 bps), is translated into a gene-product/protein with 348 residues (UniProt identifier: P08100\(^12\), SwissProt identifier: OPSD_HUMAN\(^13\)). This protein has a single large domain (Pfam identifier: PF00001) that is dominated by a “standard” 7-transmembrane receptor region (rhodopsin family), which spans most of the coding region (residues 55 - 306). The human rhodopsin is highly expressed in the heart, liver, skeletal muscle, thyroid and the eye retina.

Viewing gene expression data

It is interesting to locate the array probes intensities on the various protein domain regions. We map the expression profiles of the RHO (from GEO43134) to the structural protein regions through
visualization with the HeatMapViewer component (Figure 1). The different experimental conditions are presented on the rows, while the probes for the RHO gene are shown on the columns, annotated with exon and trans-membrane (TM) location. Probes with high expression are marked in red; those with low expression are colored green. The differences in color of the same probe along the different conditions provide useful information concerning the expression intensity of the various probes, and possible variations in alternative splicing patterns and region conservation across the different samples.

**Predicted protein mutability landscape**

Since RP is caused by mutations in the rhodopsin gene, researchers have extensively investigated different variations of the gene. Thus, up to now over 100 mutations have been identified and associated with RP. More generally: single nucleotide variations constitute most of the genetic variation among humans and therefore play an important role when studying hereditary diseases or differential drug response. In this context, we show another possible application of the HeatMapViewer, again using the 7TM human rhodopsin (SwissProt identifier: OPSD_HUMAN\(^1\)). The HeatMapViewer provides a fast and easy way to represent high dimensional data in a visually comprehensible way that immediately conveys where mutations are likely to be deleterious. Without using a tool such as the HeatMapViewer, we could hardly obtain an overview of the protein mutability landscape. Mutability landscape studies involve predicting the effect of all possible nsSNPs through computational methods, visualizing the predictions in heat maps and cross-linking these predictions with additional sources of information (such as secondary structure, active sites and correlated mutational behavior). Such regions might highlight important aspects of RP. To this end, heat maps (Figure 2b, 2c) can easily distinguish between low effect regions (represented in blue) and high effect regions (represented in red) while additional information (such as the secondary structure; Figure 2a) can simply be overlaid. These two components already perfectly convey the information that high effect regions are mainly found in the transmembrane helices and in close proximity of the binding sites. Displaying this simple fact without a heat map would be daunting due to the high dimensionality of the underlying data.

**Conclusions**

The HeatMapViewer component provides a new, powerful way to generate and display matrix data in web presentations and in publications. The use of scalable graphics enables the rendering of high-resolution images as the interactive nature of the component permits those graphics to be scaled on-demand. Furthermore the component can be applied to different cases highlighting various points of interest from gene expression levels to the effects of mutability on protein function. Finally, to make the HeatMapViewer component widely accessible, we set up a public web server to which users can upload their matrix data and use the resulting code to show an interactive heat map.

**Software availability**


**Author contributions**

GY developed the original HeatMapViewer component and maintains the HeatMapViewer service. MH contributed the SNP analysis use case. MPC and AY contributed the gene expression data use case. BR contributed the initial idea for the component and helped with its design.

**Competing interests**

No competing interests were disclosed.

**Grant information**

GY and MH are supported by a grant from the Alexander von Humboldt foundation through the German Ministry for Research
Acknowledgements
Thanks to Tim Karl and Laszlo Kajan (TUM) for invaluable help with hardware and software; to Marlena Drabik (TUM) for administrative support; to thanks Edda Kloppman and Tatyana Goldberg (TUM) for helpful discussions and invaluable feedback.

References

Open Peer Review

Current Referee Status: ✔ ✔

Version 1

Referee Report 05 March 2014

https://doi.org/10.5256/f1000research.3678.r3682

Jon C. Ison
EMBL European Bioinformatics Institute, Hinxton, Cambridge, UK

The article is very well written and tells a nice story, with compelling real-world examples. The software should have scientific impact by providing a convenient overview of matrix data, along the lines already suggested by the authors, and certainly by saving other scientist/developers the need to develop such functionality.

The software fulfills a need for an easy to use widget for heatmap rendering and there is no doubt that this is a valuable contribution to the BioJS collection. However, I believe it would help many readers if the article placed the software in the broader context of similar offerings, so it would be good if the article enumerated these, perhaps as some sort of “feature matrix”?

I appreciated the test server very much and encourage the authors to support this service in the future and further develop the functionality. Please note that examples of the CSV format are needed (at least I could not find them).

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Referee Report 27 February 2014

https://doi.org/10.5256/f1000research.3678.r3752

Jordi Deu-Pons
Biomedical Genomics Group, University Pompeu Fabra (UPF), Barcelona, Spain

This is an interesting article and piece of software. I think it contributes towards further alternatives to easily visualize high dimensionality data on the web. It’s simple and easy to embed into other web frameworks or applications.

Minor revisions

a) About the software
1 - **CSV format.** It was hard to guess the expected format. The authors need to add a syntax description of the CSV format at the help page.

2 - **Simple HTML example.** It will be easy to test HeatmapViewer (HmV) if you add a simple downloadable example file with the minimum required HTML-JavaScript to set up a HmV (without all the CSV import code).

3 - **Color scale.** HmV only implements a simple three point linear color scale. For me this is the major weakness of HmV. It will be very convenient that in the next HmV release the user can give as a parameter a function that manages the score to color conversion.

b) **About the paper**

1 - **Introduction (4th paragraph):** There are many alternatives to explore a dataset using heat maps. The author only cites two and it’s not clear if you refer to “JavaScript” or “web” alternatives. I think that you have to emphasize the strengths of HmV in comparison to other alternatives (in my opinion, one strength is that it is a good lightweight alternative to embed heat maps in a web report). Example of alternatives that I know of (but I’m sure that there are many more) are:

   - [http://www.broadinstitute.org/gsea](http://www.broadinstitute.org/gsea) (desktop)
   - [http://www.gitools.org/](http://www.gitools.org/) (desktop)
   - [http://blog.nextgenetics.net/demo/entry0044/](http://blog.nextgenetics.net/demo/entry0044/) (website)
   - [http://matplotlib.org/api/pyplot_api.html](http://matplotlib.org/api/pyplot_api.html) (python)

2 - **Predicted protein mutability landscape:** The authors say: “Without using a tool such as the HeatmapViewer, we could hardly obtain an overview of the protein mutability landscape”. This paragraph seems to suggest that you can explore the data with HmV. I think that HmV is a good tool to report your data, but not to explore it.

3 - **Conclusions:** The authors say: “...provides a new, powerful way to generate and display matrix data in web presentations and in publications.” To use heat maps in web presentations and publications is nothing new. I think that HmV makes it easier and user-friendly, but it's not new.

**Competing Interests:** No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.
The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com