SOFTWARE

shinyheatmap: ultra fast low memory heatmap software for big data genomics

Bohdan B. Khomtchouk* and Claes Wahlestedt

*Correspondence: b.khomtchouk@med.miami.edu
Center for Therapeutic Innovation and Department of Psychiatry and Behavioral Sciences, University of Miami Miller School of Medicine, 1120 NW 14th ST, Miami, FL, USA 33136
Full list of author information is available at the end of the article

Abstract

Background: Transcriptomics, metabolomics, metagenomics, and other various next-generation sequencing (-omics) fields are known for their production of large datasets. Visualizing such big data has posed technical challenges in biology, both in terms of available computational resources as well as programming acumen. Since heatmaps are used to depict high-dimensional numerical data as a colored grid of cells, efficiency and speed have often proven to be critical considerations in the process of successfully converting data into graphics. For example, rendering interactive heatmaps from large input datasets (e.g., 100k+ rows) has been computationally infeasible on both desktop computers and web browsers. In addition to memory requirements, programming skills and knowledge have frequently been barriers-to-entry for creating highly customizable heatmaps.

Results: We propose shinyheatmap: an advanced user-friendly heatmap software suite capable of efficiently creating highly customizable static and interactive biological heatmaps in a web browser. shinyheatmap is a low memory footprint program, making it particularly well-suited for the interactive visualization of extremely large datasets that cannot typically be computed in-memory due to size restrictions.

Conclusions: shinyheatmap is hosted online as a freely available web server with an intuitive graphical user interface: `http://shinyheatmap.com`. The methods are implemented in R, and are available as part of the shinyheatmap project at: `https://github.com/Bohdan-Khomtchouk/shinyheatmap`.

Background

Heatmap software can be broadly classified into two categories: static heatmap software (Saeed et al. 2003, Reich et al. 2006, Verhaak et al. 2006, Qlucore, GENE-E, Chu et al. 2008, Khomtchouk et al. 2014) and interactive heatmap software (Saldanha 2004, Caraux and Pinloche 2005, Kibbey and Calvet 2005, Wu et al. 2010, Perez-Llamas and Lopez-Bigas 2011, Škuta et al. 2014, Turkay et al. 2014, Babicki et al. 2016). Static heatmaps are pictorially frozen snapshots of genomic activity displayed as colored images generated from the underlying data. Interactive heatmaps are dynamic palettes that allow users to zoom in and out of the contents of a heatmap to investigate a specific region, cluster, or even single gene while, at the same time, being able to hover the mouse pointer over any specific row and column in order to glean information about specific cell content (e.g., gene name, expression level, and column name). Interactive heatmaps are especially important for analyzing large gene expression datasets where individual gene labels become unreadable due to text overlap, a common drawback seen in static heatmaps.

For most studies, static heatmaps are still the preferred type of publication figure, yet interactive heatmaps are becoming increasingly adopted by the scientific community to emphasize and visualize specific sectors of a dataset, where individual values are rendered as user-specified colors. As a whole, the concept of interactivity is gradually shifting the heatmap visualization field into data analytics territory, for example, by synergizing interactive (and static) heatmap software with integrated statistical and genomic analysis suites such as PCA, differential expression, gene ontology, and network analysis (Metsalu and Vilo 2015, Khomtchouk et al. 2016). However, currently existing interactive heatmap software are limited by implicit restrictions on file input size, which functionally constrains their range of utility. For example, in Clustviz (Metsalu and Vilo 2015), which employs the pheatmap R package (Kolde 2015) for heatmap generation, input datasets larger than 1000 rows are discouraged (Kolde 2015 (manual, page 4)) for performance reasons due to the computational expense associated with hierarchical clustering. Likewise, in MicroScope, the user is prompted to perform differential expression analysis on the input dataset first, thereby shrinking the number of rows rendered in the interactive heatmap to encompass only statistically significant genes (Khomtchouk et al. 2016). In general, the standard way of thinking has been to entirely avoid the production of big heatmaps due primarily to poor readability and unclear interpretation (Google Groups 2012, SO 2013, Mango Information Systems 2013, SO 2014, vida.io 2014). However, with the advent of increasingly sophisticated interactive heatmap software and the rise of big data coupled with a growing community interest to examine it interactively, there has arisen an unmet and pressing need to address the computational limitations that hinder the production of large, interactive heatmaps. To this end, we propose an ultra fast and low memory user-friendly heatmap software suite capable of efficiently creating highly customizable static and interactive heatmaps in a web browser.

Implementation

shinyheatmap is hosted online as an R Shiny web server application. shinyheatmap may also be run locally from within R Studio, as shown here: `https://github.com/Bohdan-Khomtchouk/shinyheatmap`. shinyheatmap leverages the cumulative utility of R's heatmaply (Galili 2016), shiny (Chang et al. 2015), data.table (Dowle et al. 2015), and gplots (Warnes et al. 2016) libraries to create a cohesive web browser-based software experience requiring absolutely no programming experience from the user, or even the need to download R on a local computer.

For the static heatmap generation, shinyheatmap employs the heatmap.2 function of the gplots library. For the interactive heatmap generation, shinyheatmap employs the heatmaply R package, which directly calls the plotly.js engine, in order to create fast, interactive heatmaps from large input datasets. The heatmaply R package is a descendent of the d3heatmap R package, which successfully creates advanced interactive heatmaps but is incapable of handling large inputs (e.g., 20000+ rows) due to memory considerations. As such, heatmaply constitutes a much-needed performance upgrade to d3heatmap, one that is made possible by the plotly R package (Sievert et al. 2016), which itself relies on the very sophisticated and complex plotly.js engine (Plotly Technologies Inc. 2016). Therefore, it is the technical innovations of the plotly.js source code that make drawing extremely large heatmaps both a fast and efficient process. However, heatmaply adds certain features not present in either the plotly.js engine nor the plotly R package, namely the ability to perform advanced hierarchical clustering and dendrogram-side zooming.

Results

To use shinyheatmap, input data must be in the form of a matrix of integer values. The value in the i-th row and the j-th column of the matrix tells how many reads (or fragments, for paired-end RNA-seq) have been unambiguously assigned to gene i in sample j (Love et al. 2016). Analogously, for other types of assays, the rows of the matrix might correspond e.g., to binding regions (with ChIP-seq), species of bacteria (with metagenomic datasets), or peptide sequences (with quantitative mass spectrometry). For detailed usage considerations, shinyheatmap provides a convenient Instructions tab panel upon login.

Upon uploading such a dataset, both static and interactive heatmaps are automatically created, each in their own respective tab panel. The user can then proceed to customize the static heatmap through a suite of available parameter settings located in the sidebar panel (Figure 1). For example, hierarchical clustering, color schemes, scaling, color keys, trace, and font size can all be set to the specifications of the user. In addition, a download button is provided for users to save publication quality heatmap figures. Likewise, the user can customize the interactive heatmap through its own respective panel located at the upper right corner of the heatmap (Figure 2). This panel provides extensive download, zoom, pan, lasso and box select, autoscale, reset, and hover features for interacting with the heatmap.

Conclusion

We provide access to a user-friendly web application designed to quickly and efficiently create static and interactive heatmaps within the R programming environment, without any prerequisite programming skills required of the user. Our software tool aims to enrich the genomic data exploration experience by providing a variety of customization options to investigate large input datasets.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author's contributions

BBK conceived the study and wrote the code. CW participated in the management of the source code and its coordination. BBK wrote the paper. All authors read and approved the final manuscript.

Acknowledgements

BBK wishes to acknowledge the financial support of the United States Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program: this research was conducted with Government support under and awarded by DoD, Army Research Office (ARO), National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. Relevant work in CW's laboratory is currently funded by NIH grants DA035592 and AA023781.

References

1. Babicki S, Arndt D, Marcu A, Liang Y, Grant JR, Maciejewski A, Wishart DS: *Heatmapper: web-enabled heat mapping for all*. Nucleic Acids Research 2016, pii: gkw419. [Epub ahead of print].
2. Caraux G, Pinloche S: *Permutmatrix: A Graphical Environment to Arrange Gene Expression Profiles in Optimal Linear Order*. Bioinformatics. 2005, 21: 1280–1281.

3. Chang W, Cheng J, Allaire JJ, Xie Y, McPherson J, RStudio, jQuery Foundation, jQuery contributors, jQuery UI contributors, Otto M, Thornton J, Bootstrap contributors, Twitter Inc, Farkas A, Jehl S, Petre S, Rowls A, Gandy D, Reavis B, Kowal KM, es5-shim contributors, Ineshin D, Samhuri S, SpryMedia Limited, Fraser J, Gruber J, Sagalaev I, R Core Team: *shiny: Web Application Framework for R*. 2015. R package version 0.12.2.
4. Cheng J, Galili T, RStudio Inc, Bostock M, Palmer J: *d3heatmap: Interactive Heat Maps Using 'htmlwidgets' and 'D3.js'*. 2015. R package version 0.6.1.
5. Chu VT, Gottardo R, Raftery AE, Bumgarner RE, Yeung KY: *MeV+R: using MeV as a graphical user interface for Bioconductor applications in microarray analysis*. Genome Biology. 2008, 9: R118.
6. Dowle M, Srinivasan A, Short T, Lianoglou S, Saporta R, Antonyan E: *data.table: Extension of Data.frame*. 2015. R package version 1.9.6.
7. Galili T: *heatmaply: Interactive Heat Maps Using 'plotly'*. 2016. R package version 0.5.0. https://CRAN.R-project.org/package=heatmaply.
8. Gould J: GENE-E software hosted at the Broad Institute. http://www.broadinstitute.org/cancer/software/GENE-E/.
9. Khomtchouk BB, Van Booven DJ, Wahlestedt C: *HeatmapGenerator: high performance RNAseq and microarray visualization software suite to examine differential gene expression levels using an R and C++ hybrid computational pipeline*. Source Code for Biology and Medicine. 2014, 9(1): 1–6.
10. Khomtchouk BB, Hennessy JR, Wahlestedt C: *MicroScope: ChIP-seq and RNA-seq software analysis suite for gene expression heatmaps*. BMC Bioinformatics. 2016, in press. doi:http://dx.doi.org/10.1101/034694
11. Kibbey C, Calvet A: *Molecular Property eXplorer: a novel approach to visualizing SAR using tree-maps and heatmaps*. J Chem Inf Model. 2005, 45(2): 523–532.
12. Kolde R: *pheatmap: Pretty Heatmaps*. 2015. R package version 1.0.8. https://CRAN.R-project.org/package=pheatmap.
13. Kolde R: *pheatmap: Pretty Heatmaps*. 2015. Page 4: ftp://cran.r-project.org/pub/R/web/packages/pheatmap/pheatmap.pdf.
14. Love M, Anders S, Kim V, Huber W: *RNA-seq workflow: gene-level exploratory analysis and differential expression*. 2016, http://www.bioconductor.org/help/workflows/rnaseqGene/.
15. Metsalu T, Vilo J: *ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap*. Nucleic Acids Research. 2015, 43(W1): W566–570.
16. Perez-Llamas C, Lopez-Bigas N: *Gitools: analysis and visualisation of genomic data using interactive heat-maps*. PLoS One. 2011, 6: e19541.
17. Plotly Technologies Inc.: *Collaborative data science*. Plotly Technologies Inc. Montreal, QC. 2015, https://plot.ly.
18. Qlucore Omics Explorer: The D.I.Y Bioinformatics Software. http://www.qlucore.com.
19. R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
20. Reich M, Liefeld T, Gould J, Lerner J, Tamayo P, Mesirov JP: *GenePattern 2.0*. Nat Genet. 2006, 38(5): 500–501. 10.1038/ng0506-500.
21. Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M, Sturn A, Snuffin M, Rezantsev A, Popov D, Ryltsov A, Kostukovich E, Borisovsky I, Liu Z, Vinsavich A, Trush V, Quackenbush J: *TM4: a free, open-source system for microarray data management and analysis*. Biotechniques. 2003, 34(2): 374–378.
22. Saldanha AJ: *Java Treeview – extensive visualization of microarray data*. Bioinformatics. 2004, 20(17): 3246–3248.
23. Sievert C, Parmer C, Hocking T, Chamberlain S, Ram K, Corvellec M, Despouy P: *plotly: Create Interactive Web Graphics via 'plotly.js'*. 2016. R package version 3.6.0. https://CRAN.R-project.org/package=plotly.
24. Škuta C, Bartůněk P, Svozil D: *InCHlib — interactive cluster heatmap for web applications* Journal of Cheminformatics. 2014, 6(44): 1–9.
25. SO, 2014. How to draw heatmap with huge data. http://stackoverflow.com/questions/23297616/how-to-draw-heatmap-with-huge-data.
26. SO, 2013. D3: How to show large dataset. http://stackoverflow.com/questions/18244995/d3-how-to-show-large-dataset
27. Google Groups, 2012. Heat map with 500*300 nodes. https://groups.google.com/forum/m/#!topic/d3-js/wVWvwa-YkFE
28. Mango Information Systems, 2013. Pre-render d3.js charts at server side. https://mango-is.com/blog/engineering/pre-render-d3-js-charts-at-server-side/
29. vida.io, 2014. BigQuery Big Data Visualization With D3.js. http://blog.vida.io/2014/07/06/bigquery-big-data-visualization-with-d3-dot-js/
30. Tan MH, Au KF, Yablonovitch AL, Wills AE, Chuang J, Baker JC, Wong WH, Li JB: *RNA sequencing reveals a diverse and dynamic repertoire of the Xenopus tropicalis transcriptome over development*. Genome Research. 2013, 23: 201–216.
31. Turkay C, Lex A, Streit M, Pfister H, Hauser H: *Characterizing cancer subtypes using dual analysis in Caleydo StratomeX*. IEEE Comput Graph Appl. 2014, 34(2): 38–47.
32. Verhaak RGW, Sanders MA, Bijl MA, Delwel R, Horsman S, Moorhouse MJ, van der Spek PJ, Lowenberg B, Valk PJM: *HeatMapper: powerful combined visualization of gene expression profile correlations, genotypes, phenotypes and sample characteristics*. BMC Bioinformatics. 2006, 7:337.
33. Warnes GR, Bolker B, Bonebakker L, Gentleman R, Huber W, Liaw A, Lumley T, Maechler M, Magnusson A, Moeller S, Schwartz M, Venables B: *gplots: Various R Programming Tools for Plotting Data*. 2016. R package version 3.0.1. https://CRAN.R-project.org/package=gplots.
34. Wu HM, Tien YJ, Chen CH: *GAP: A Graphical Environment for Matrix Visualization and Cluster Analysis*. Computational Statistics and Data Analysis. 2010, 54: 767–778.

Figures

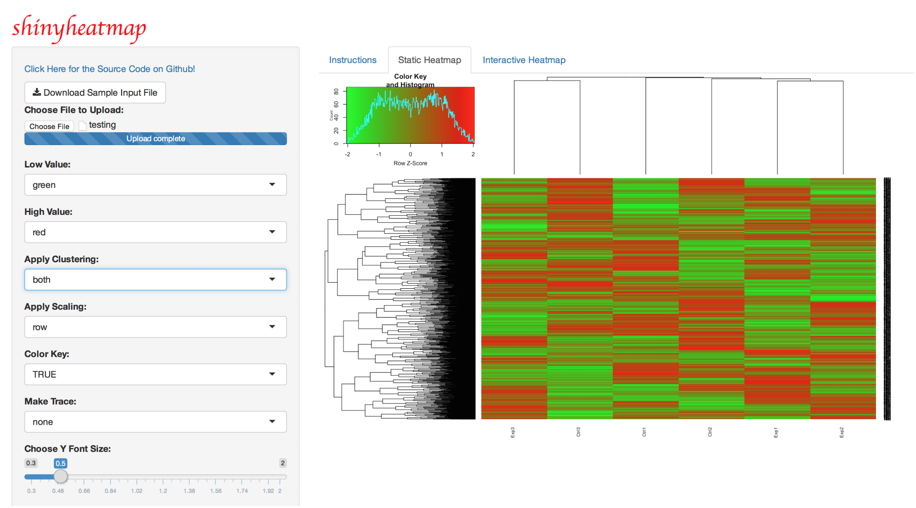


Figure 1 shinyheatmap static heatmap. shinyheatmap UI showcasing the visualization of a static heatmap generated from a large input dataset. Parameters such as hierarchical clustering, color schemes, scaling, color keys, trace, and font size can all be set by the user. A sample input file is provided as part of the web application, whose source code can be viewed on Github.

Figure 2 shinyheatmap interactive heatmap. shinyheatmap UI showcasing the visualization of an interactive heatmap generated from a large input dataset. An embedded panel provides extensive download, zoom, pan, lasso and box select, autoscale, reset, and hover features for interacting with the heatmap.

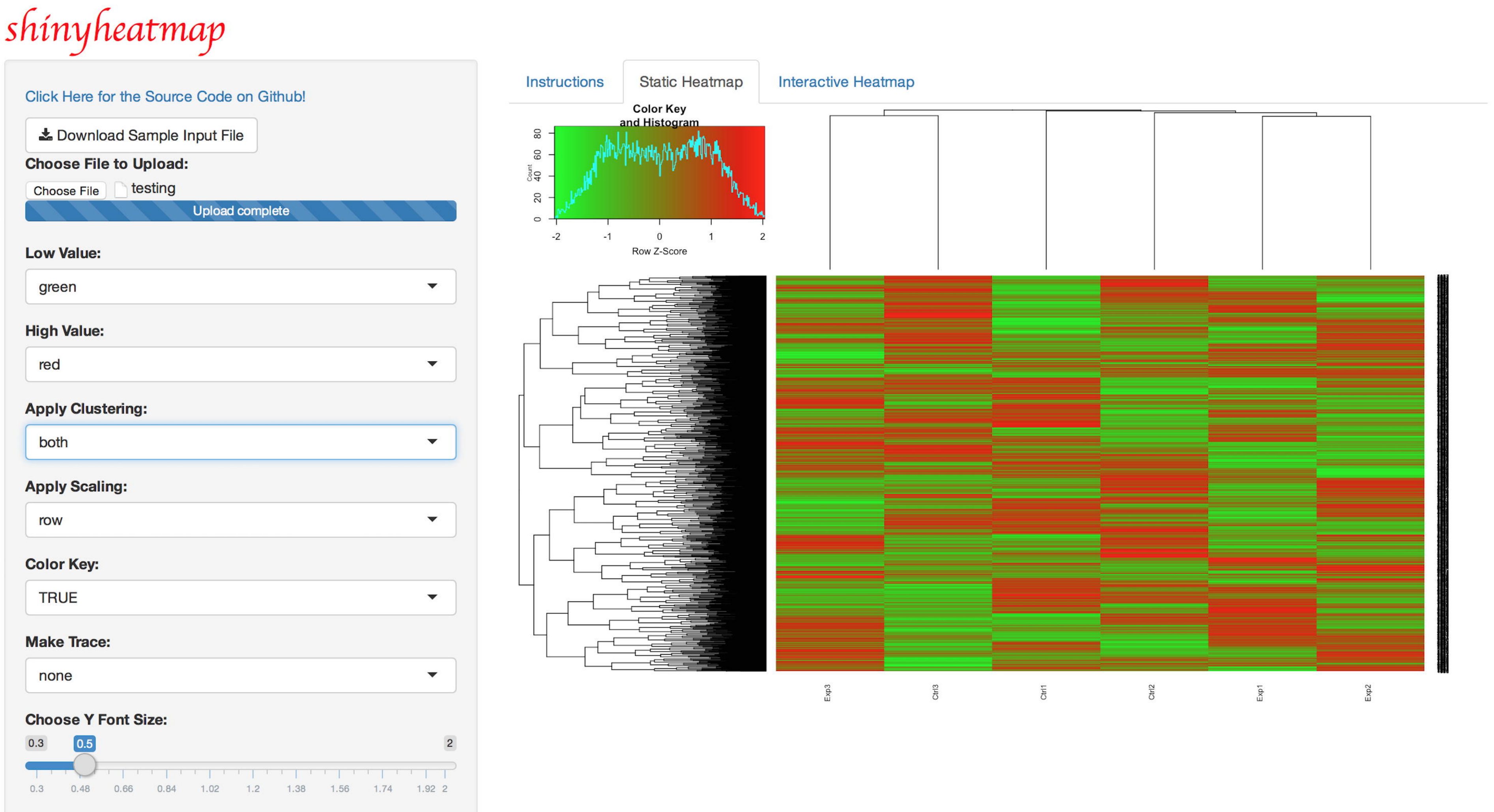
Ethics and consent to participate
This study does not involve humans, human data or animals.

Consent to publish
Not applicable.

Funding
Funding provided to BBK from the United States Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program: this research was conducted with Government support under and awarded by DoD, Army Research Office (ARO), National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. Relevant work in CW's laboratory is currently funded by NIH grants DA035592 and AA023781.

Abbreviations used
PCA: principal component analysis
UI: user interface

Availability of Data and Materials
All source code has been made publicly available on Github at:
https://github.com/Bohdan-Khomtchouk/shinyheatmap.


shinyheatmap
Click Here for the Source Code on Github!
Download Sample Input File
Choose File to Upload:
Choose File
testing
Upload complete
Low Value:
green
High Value:
red
Apply Clustering:
both
Apply Scaling:
row
Color Key:
TRUE
Make Trace:
none
Choose Y Font Size:
0.3
0.5
2
0.3 0.48 0.66 0.84 1.02 1.2 1.38 1.56 1.74 1.92 2
Instructions
Static Heatmap
Interactive Heatmap
Color Key
and Histogram
Count
0 20 40 60 80
-2 -1 0 1 2
Row Z-Score
Exp3
Ctrl3
Ctrl1
Ctrl2
Exp1
Exp2


shinyheatmap
Click Here for the Source Code on Github!
Download Sample Input File
Choose File to Upload:
Choose File
testing
Upload complete
Low Value:
green
High Value:
red
Apply Clustering:
none
Apply Scaling:
row
Color Key:
TRUE
Make Trace:
none
Choose Y Font Size:
0.3
0.5
2
0.3 0.48 0.66 0.84 1.02 1.2 1.38 1.56 1.74 1.92 2
Instructions
Static Heatmap
Interactive Heatmap
column: Ctrl1
row: 807
value: 449
5000
4000
3000
2000
1000
0
Exp3
Ctrl3
Ctrl2
Exp1
Exp2
Ctrl1