

SOFTWARE

bigPint: A Bioconductor package that makes big data pint-sized

Lindsay Rutter^{1*} and Dianne Cook²

*Correspondence:

lindsayannerutter@gmail.com

¹ Bioinformatics andComputational Biology Program,
Iowa State University, Ames, USAFull list of author information is
available at the end of the article**Abstract****Keywords:** sample; article; author**Background**

Interactive data visualization is increasingly imperative in the biological sciences [1]. When performing RNA-seq studies, researchers wish to determine which genes are differentially expressed between treatment groups. Interactive visualization can help them assess differentially expressed gene (DEG) calls before performing any subsequent functional enrichment analyses. New visualization tools for genomic data have incorporated interactive capabilities, and some believe this trend could enhance the exploration of genomic data in the future [2]. Despite the growing appreciation of the inherent value of interactive graphics, the availability of effective and easy-to-use interactive visualization tools for RNA-seq data remains limited.

Some interactive visualization tools for genomic data have limited usability because they are only available on certain operating systems and/or require payment [3, 4, 5]. These limitations can be removed when tools are published on open-source repositories. Indeed, the Bioconductor project aims to foster interdisciplinary scientific research by promoting transparency and reproducibility while allowing software content to be used on Windows, MacOS, and Linux [6]. Bioconductor software is written in the R programming language, which also provides statistical and visualization methods that can facilitate the development of robust graphical tools in computational biology. Several interactive visualization methods for genomic data have been developed using Shiny, which is also based on R programming language [7, 8, 9].

We recently developed bigPint, an interactive data visualization software package available on Bioconductor. The bigPint package allows users to visually explore many types of large multivariate datasets, even though it was more specifically developed for RNA-seq data. In a recent methods paper, we used public RNA-seq datasets to demonstrate how bigPint graphics can help biologists detect crucial issues with normalization methods and DEG designation in ways not possible with numerical models [10]. We also applied bigPint visualization tools in a recent research paper that sought to elicit how nutrition and viral infection affect the honey bee transcriptome [11]. In the current paper, we explain the technical innovations and merits of the bigPint package, including new interactive visualization techniques that we believe can be helpful in the development and

usage of future biological visualization software. The bigPint website is located at <https://bioconductor.org/packages/devel/bioc/html/bigPint.html> and contains short vignette articles that introduce new users and provide suggested analysis pipelines, all written in reproducible code.

Results

Basic input

Each method in bigPint requires an input parameter data object. If a researcher is using the package to visualize RNA-seq data, then this data object should be a count table that contains the read counts for all genes of interest. The value in row i and column j should indicate how many reads have been assigned to gene i in sample j . This is the same input format required in popular RNA-seq count-based statistical methods, such as DESeq2, edgeR, limma, DSS, EBSeq, and BaySeq [?].

Several methods in bigPint also require an input parameter dataMetrics object. If a researcher is using the package to visualize RNA-seq data, then this dataMetrics object should be subset of the data (usually DEGs) where each case includes quantitative values of interest (such as fold change and FDR). This information can be easily derived from popular RNA-seq numerical analysis packages. Again, this conveniently allows users to work smoothly between visualizations in the bigPint package and models in other Bioconductor packages, abiding to the notion that the most efficient way to analyze large datasets is to iterate between models and visualizations.

Original features

Separate layers of dynamics and interactivity

The Bioconductor community advanced the boundaries of biological visualization in the past and believes that interactive technology must be incorporated to continue these advancements in the future [6]. We developed what we believe is a fairly new interactive visualization technology that could be useful in the exploration of many types of large biological datasets. Our technique allows for two separate layers of interactive and dynamic capabilities, one for the foreground of the plot and one for the background of the plot. With interaction, users can interact with displayed data. This could mean hovering over a graphical representation of the data (such as a hexagon) to obtain metadata about the elements within that graphical representation (such as the names of its contained genes).

This is especially true with large RNA-seq datasets, where the background can contain the entire dataset and the foreground can contain only a subset of the dataset (such as one or more DEGs). A user may wish to overlay one DEG in the foreground across the entire dataset in the background

1. Scatterplot matrices

Scatterplot matrices have appeared in statistical graphics literature for almost four decades and used across various fields of research [12, 13, 14, 15]. Previous user studies have shown that participants performed better when using animated rather than static versions of scatterplot matrices. Users also preferred animated scatterplot matrices and found them easier to understand as they can alleviate overplotting issues

[16]. Interaction has been shown to extend the scatterplot matrix into an effective tool when representing large datasets [17]. We believe our new interactive visualization technology may further improve upon this long-standing plotting technique known for its effectiveness in exploratory multivariate data analysis .

2. Litre plots

Several problems can still remain when scatterplot matrices are applied to large datasets. Physical space requirements grow exponentially by dimension size: for n -dimensional data, n^2 scatterplots are typically drawn. Hence, when scatterplot matrices are extended to large dimensions, it can become difficult to mentally link the many small plots in the matrix [18].

Useful features

Tailoring and saving static plots

Static plots can be saved as list objects in the R workspace and/or as JPG files to a directory chosen by the user. Saving plots into the R workspace allows users to integrate them into analysis workflows. It also allows them to tailor the plots (such as adding titles and changing label sizes) using the grammar of graphics via the conventional `+` syntax. Saving plots to a directory allows users to keep professional-looking files that can be inserted into proposals and talks. By default, the `bigPint` package saves static plots both in the R workspace and a directory (the default location is `tempdir()`).

Second feature layer

All plots, but static and interactive, allow for a selection of data to be plotted in a different way than the rest of the data. When analyzing RNA-seq data, this second feature layer could represent DEGs. There are three options for superimposing data subsets. First, users can declare data subsets using the previously-mentioned `dataMetrics` object by thresholding one of its quantitative variables. Second, users can simply declare a list of IDs to be overlaid using an object called `geneList`. This allows for more flexibility than the first option. Third, the user can simply not overlay data subsets by leaving the `dataMetrics` and `geneList` objects with a value of `NULL`.

Group comparison filters

When users create static plots, the package automatically creates a separate plot for each pairwise combination of treatment groups from the inputted data. When users explore interactive plots, fields are dynamically generated from the inputted data so that any pairwise combination of treatment groups can be selected. Users can then quickly flip between different contrasts in their data. The package comes with an example dataset of soybean cotyledon data that has three treatment groups and examples of these contrast functionalities are shown on this data throughout the easy-to-follow articles on the package website.

Hexagonal binning

Users can create scatterplot matrices and litre plots using points (where each point represents one gene) or hexagonal binnings (where each hexagon color represents the number of genes in that area). Plotting each gene as a point allows for ideal levels of detail until overplotting occurs, which makes it difficult to determine how many genes are in given areas. In large datasets, this problem often persists even with techniques like alpha blending. Hexagonal binning has been used in prior software to successfully manage overplotting issues [17, 19] and has shown superior time performance because less geometric objects need to be plotted. The bigPint package allows users to plot using either raw points or hexagonal binning, either of which can be useful depending on the dataset.

Hierarchical clustering

Users can conduct hierarchical clustering analyses on their data using the function `plotClusters()`. By default, the resulting clusters will be plotted as parallel coordinate lines superimposed onto side-by-side boxplots that represent the distribution of the full dataset. There are three main approaches in the `plotClusters()` function:

- Approach 1: The clusters are formed by clustering only on a user-defined subset of data (such as significant genes). Only these user-defined genes are overlaid as parallel coordinate lines.
- Approach 2: The clusters are formed by clustering the full dataset. Then, only a user-defined subset of data (such as significant genes) are overlaid as parallel coordinate lines.
- Approach 3: The clusters are formed by clustering the full dataset. All genes are overlaid as parallel coordinate lines.

The clustering algorithm is based on the `hclust()` and `cutree()` functions in the R stats package. It offers the same set of agglomeration methods (“ward.D”, “ward.D2”, “single”, “complete”, “average”, “mcquitty”, “median”, and “centroid”) with “ward.D” as the default. In many cases, users may want to save clusters derived from the `plotClusters()` function for later use, such as to overlay them onto scatterplot matrices, litre plots, and volcano plots. The gene IDs of each cluster can be saved as .RDS files for this purpose by setting the `verbose` option of the `plotClusters()` function to a value of `TRUE`.

Various plot aesthetics

Users can modify various aesthetics for both static and interactive plots, including hexagon size and point size. Some plots also allow for alpha blending, which has proven beneficial when plotting certain large datasets into parallel coordinate lines [20]. Statistical coloring is inconsistent in many numerous packages even though it can greatly enhance biological data visualization [21]. The bigPint package allows users to easily maintain consistent coloring across hierarchical clusters and when working between various plots.

Selection and aggregation

Some techniques that are effective in data exploration may lose their efficiency and eventually fail as data items increase in size. Two main approaches to solving these

problems include data selection and data aggregation [22]. Data selection means that only a subset of the full dataset is displayed at a given time. The data subset can be selected through queries where data items are only displayed if they meet certain requirements [22]. The data subset can also be selected through interactive controls which allow the user to quickly examine between various subsets of the data [22]. Data aggregation means that the full dataset is divided into data subsets (called aggregates) that reduce the amount of data being visualized at once. When working with large datasets, an appropriate compromise is for users to be able to perform both data selection and data aggregation [22]. The bigPint package allows users to easily perform data selection using queries and interactive controls (such as zooming, box and lasso selection, and panning) and to perform data aggregation using hierarchical clustering.

Shiny interactivity

Interactive plots in the bigPint package open as Shiny applications that consist of simple dashboards with “About” tabs that explain how to use the applications. They also include “Application” tabs that provide several input fields for the user to tailor their plots. Some of these input fields are generated dynamically from the inputted dataset so users have more flexibility in how they select any data subsets they wish to superimpose. In these applications, users can also download lists of selected genes and static images of interactive graphics to their local computers.

Shiny allows for linking between plots. Linking functionality between these plots plays a crucial role in rendering them suitable for large datasets [23, 24, 25]. By combining Shiny with Plotly and htmlwidgets functions, the bigPint package offers novel ways of dynamically and interactively working within and between plots.

Shiny applications can be launched on a local personal computer, hosted on a local or cloud-based server, or hosted for free on the shinyapps.io website. As such, interactive bigPint packages can be deployed on a personal computer using only a local file containing the data, the bigPint package and its dependencies, R / RStudio, and a browser recommended by Shiny (Google Chrome or Mozilla Firefox). This method does not require internet connectivity, which can be useful for users who are protecting sensitive data, analyzing or presenting data in contexts without reliable connectivity, or testing and developing applications.

Discussion

Researchers benefit when they are able to view multiple perspective of their data, especially when working with large datasets [26, 27]. The ability to select and aggregate data, threshold data to create subsets, link between multiple plots, interact with plots, and tailor various aesthetics in intelligent ways are all useful features of the bigPint package [1, 2, 18]. We expect that bigPint will enable researchers to generate and interact with intuitive, high quality, and reproducible plots from increasingly large datasets, including RNA-seq datasets.

Conclusion

Despite the growing appreciation of the inherent value in interactive graphics, the availability of easy-to-use and effective interactive exploratory visualization tools

for RNA-seq data remains limited. In this paper, we introduced new visualization techniques with dynamic and interactive capabilities that are separately layered in the foreground and background of plots. We believe this methodology represents a fairly novel contribution to the field of interactive data visualization. Advocating state-of-the-art visualization tools is crucial for biology researchers to analyze and present their data and for visualization researchers to develop novel methods. Lessons learned from our open-source work may encourage the development of additional interactive visualization tools for various computational tasks.

Methods

bigPint was released under the GPL-3 license. Most bigPint visualization methods were constructed using `htmlwidgets` [28], `ggplot2` [29], `shiny` [30], `shinyapps.io` [31], and `plotly` [32]. bigPint methods were tested on numerous RNA-seq datasets [10, 11]. The package website was constructed using the `pkgdown` software [33]. bigPint can be downloaded from the Bioconductor website [6].

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Text for this section ...

Acknowledgements

Text for this section ...

Author details

¹ Bioinformatics and Computational Biology Program, Iowa State University, Ames, USA. ² Econometrics and Business Statistics, Monash University, Clayton VIC, Australia.

References

- O'Donoghue, S.I., Gavin, A.-C., Gehlenborg, N., Goodsell, D.S., Hériché, J.-K., Nielsen, C.B., North, C., Olson, A.J., Procter, J.B., Shattuck, D.W., *et al.*: Visualizing biological data—now and in the future. *Nature methods* **7**(3), 2 (2010)
- Pavlopoulos, G.A., Malliarakis, D., Papanikolaou, N., Theodosiou, T., Enright, A.J., Iliopoulos, I.: Visualizing genome and systems biology: technologies, tools, implementation techniques and trends, past, present and future. *Gigascience* **4**(1), 38 (2015)
- Seo, J., Shneiderman, B.: Interactively exploring hierarchical clustering results [gene identification]. *Computer* **35**(7), 80–86 (2002)
- Ahlberg, C.: Spotfire: an information exploration environment. *ACM SIGMOD Record* **25**(4), 25–29 (1996)
- Chu, L., Scharf, E., Kondo, T.: Genespringtm: tools for analyzing microarray expression data. *Genome Informatics* **12**, 227–229 (2001)
- Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., *et al.*: Bioconductor: open software development for computational biology and bioinformatics. *Genome biology* **5**(10), 80 (2004)
- Rue-Albrecht, K., Marini, F., Soneson, C., Lun, A.T.: isee: Interactive summarizedexperiment explorer. *F1000Research* **7** (2018)
- Schultheis, H., Kuenne, C., Preussner, J., Wiegandt, R., Fust, A., Bentsen, M., Looso, M.: Wilson: Web-based interactive omics visualization. *Bioinformatics* **35**(6), 1055–1057 (2018)
- Hughes, L.D., Lewis, S.A., Hughes, M.E.: Expressiondb: An open source platform for distributing genome-scale datasets. *PloS one* **12**(11), 0187457 (2017)
- Rutter, L., Moran-Lauter, A.N., Graham, M.A., Cook, D.: Visualization methods for rna-sequencing data analysis. Submitted to *BMC Bioinformatics*
- Rutter, L., Carrillo-Tripp, J., Bonning, B.C., Cook, D., Toth, A.L., Dolezal, A.G.: Transcriptomic responses to diet quality and viral infection in *apis mellifera*. Submitted to *BMC Genomics*
- Becker, R.A., Cleveland, W.S.: Brushing a scatterplot matrix: High-interaction graphical methods for analyzing multidimensional data. submitted for publication (1984)
- Carr, D., Nicholson, W.: Graphical interaction tools for multiple 2-and 3-dimensional scatterplots. Technical report, Pacific Northwest Lab., Richland, WA (USA) (1984)
- Tufte, E.R.: *The Visual Display of Quantitative Information* vol. 2. Graphics press Cheshire, CT, ??? (2001)
- Tukey, P., Tukey, J.: Graphical display of data sets in three or more dimensions. Three papers in *Interpreting Multivariate Data* (ed. V. Barnett), 189–275. Chichester: Wiley (1981)
- Chen, H., Engle, S., Joshi, A., Ragan, E.D., Yuksel, B.F., Harrison, L.: Using animation to alleviate overdraw in multiclass scatterplot matrices. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 417 (2018). ACM

17. Carr, D.B., Littlefield, R.J., Nicholson, W., Littlefield, J.: Scatterplot matrix techniques for large n. *Journal of the American Statistical Association* **82**(398), 424–436 (1987)
18. Kerren, A., Ebert, A., Meyer, J.: *Human-Centered Visualization Environments: GI-Dagstuhl Research Seminar, Dagstuhl Castle, Germany, March 5-8, 2006, Revised Papers vol. 4417*. Springer, ??? (2007)
19. Harshbarger, J., Kratz, A., Carninci, P.: Deiva: a web application for interactive visual analysis of differential gene expression profiles. *BMC genomics* **18**(1), 47 (2017)
20. Unwin, A., Chen, C.-h., Härdle, W.: *Computational Statistics and Data Visualization*. Humboldt-Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät, ??? (2007)
21. Yin, T., Cook, D., Lawrence, M.: ggbio: an r package for extending the grammar of graphics for genomic data. *Genome biology* **13**(8), 77 (2012)
22. Andrienko, G., Andrienko, N.: Blending aggregation and selection: Adapting parallel coordinates for the visualization of large datasets. *The Cartographic Journal* **42**(1), 49–60 (2005)
23. Nguyen, Q.V., Simoff, S., Qian, Y., Huang, M.L.: Deep exploration of multidimensional data with linkable scatterplots. In: *Proceedings of the 9th International Symposium on Visual Information Communication and Interaction*, pp. 43–50 (2016). ACM
24. Albuquerque, G., Eisemann, M., Lehmann, D.J., Theisel, H., Magnor, M.A.: Quality-based visualization matrices. In: *VMV*, pp. 341–350 (2009)
25. Heinrich, J., Stasko, J., Weiskopf, D.: The parallel coordinates matrix. *EuroVis-Short Papers*, 37–41 (2012)
26. Swayne, D.F., Lang, D.T., Buja, A., Cook, D.: Ggobi: evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis* **43**(4), 423–444 (2003)
27. Cook, D., Swayne, D.F., Buja, A.: *Interactive and Dynamic Graphics for Data Analysis: with R and GGobi*. Springer, ??? (2007)
28. Vaidyanathan, R., Xie, Y., Allaire, J., Cheng, J., Russell, K.: *Htmlwidgets: HTML Widgets for R*, 2016. <https://cran.r-project.org/package=htmlwidgets> Accessed 2018-12-20
29. Wickham, H.: *Ggplot2: Elegant Graphics for Data Analysis*. Springer, ??? (2016)
30. Chang, W., Cheng, J., Allaire, J., Xie, Y., McPherson, J.: *Shiny: Web Application Framework for R [Computer Software]*. <https://cran.r-project.org/package=shiny> Accessed 2018-12-20
31. RStudio: *Integrated Development for R*. <http://www.rstudio.com> Accessed 2018-12-20
32. Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., Despouy, P.: *Plotly: Create Interactive Web Graphics Via 'plotly.js'*. <https://cran.r-project.org/package=plotly> Accessed 2018-12-20
33. Wickham, H., Hesselberth, J.: *Pkgdown: Make Static HTML Documentation for a Package*. <https://cran.r-project.org/package=pkgdown> Accessed 2018-12-20

Figures

Figure 1 Sample figure title. A short description of the figure content should go here.

Figure 2 Sample figure title. Figure legend text.

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title

Additional file descriptions text.