The Journal of Open Source Software

visxhclust: An R Shiny package for visual exploration of hierarchical clustering

Rafael Henkin[1] **and Michael R. Barnes**[1]

1 Centre for Translational Bioinformatics, William Harvey Research Institute, Faculty of Medicine and Dentistry, Queen Mary University of London

DOI: 10.21105/joss.04074

Software

- Review
- Repository
- Archive

Editor: Øystein Sørensen

Reviewers:

- @jonjoncardoso
- @wiljnich

Submitted: 11 January 2022
Published: 02 February 2022

Summary

visxhclust is an R Shiny (Chang et al., 2021) web app that facilitates iterative exploration of hierarchical clustering. The package assembles together the outputs of different steps of a clustering workflow in a fluid visual analytics (Keim et al., 2008) interface, enabling analysts to change selected features and hyperparameters and quickly re-evaluate results. It focuses on hierarchical clustering, a widely used method that is often chosen due to the familiarity of analysts with the tree-like structure of results presented as dendrograms and its frequent combination with visualization techniques such as heatmaps. Package reference and an illustrated tutorial are available online.

Statement of need

Clustering methods are supported by dozens of packages in popular data analysis languages such as R and Python. In R, basic clustering functions are included with standard R installations (R Core Team, 2021) and are extended by many packages (e.g. `dendextend` (Galili, 2015) to modify dendrograms). In Python, state-of-the-art packages such as scikit-learn (Pedregosa et al., 2011) also include considerable support for clustering. In the R ecosystem, packages such as Radiant (Nijs, 2021) and ClustVis (Metsalu & Vilo, 2015) contain Shiny apps that include hierarchical clustering and heatmaps. The first includes a limited set of plots of clustering results and evaluation, while the second has a stronger focus on the preprocessing steps with dimensionality reduction using PCA and does not include evaluation or analysis of the computed clusters.

The visxhclust package addresses the overheads and difficulties that emerge when analysts have to combine multiple packages to explore different configurations of hyperparameters and features and additionally evaluate and interpret results. It integrates computation, plotting and evaluation of hierarchical clustering in a visual analytics interface by building on state-of-the-art R packages. The interface enables fast iterative workflows without requiring users to learn how to use specific packages or manipulate inputs and outputs of different packages. While the web app does not require any programming knowledge at all to be used, the package also exports internal functions for reproducibility and preparation of figures for publication.

Overview

Users can install the package in their local R installation and run the Shiny app off their IDE or command-line interface. The app layout is split into a sidebar, through which the user can

load data and change the clustering settings, and a main view where the user can navigate through tabs with various outputs (see Figure 1). Once the app is running, users can begin the clustering step of their analytical workflow by loading the data. By default, the tool will transform all features into standard scores by using the `scale()` function from R, though users can load standardized datasets and disable the scaling function. The initial setup will compute two clusters using Euclidean distance and Ward's linkage method; with these settings, when the user loads a dataset, almost every clustering-related output will be updated across the app as soon as the user opens the corresponding tab. The tool supports the distance measures included in standard R distributions (Euclidean, maximum, Manhattan, Canberra, binary and Minkowski), plus cosine and Mahalanobis distance (Mahalanobis, 1936).

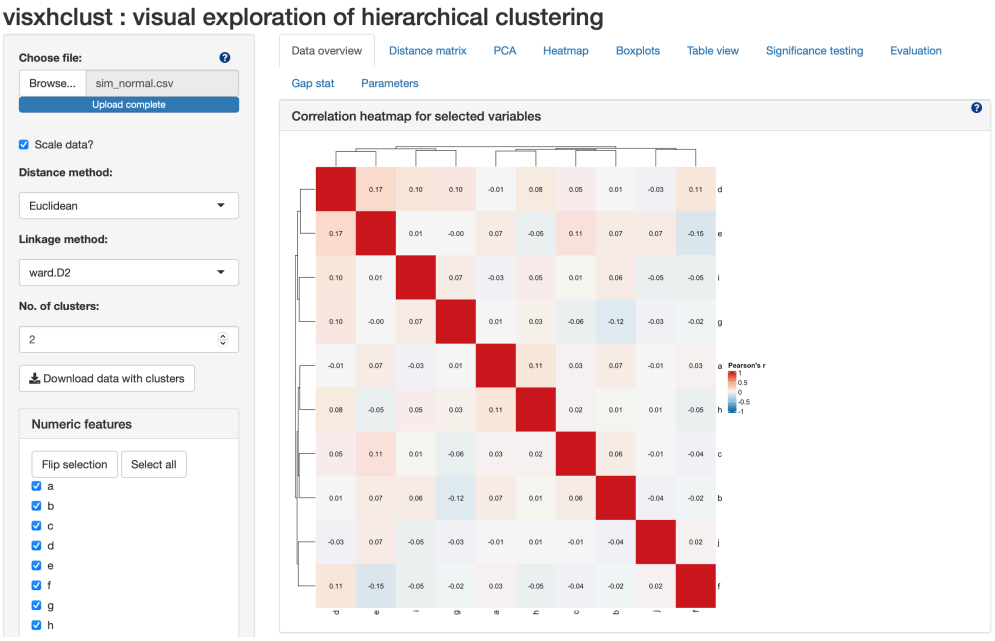

Figure 1: Interface of visxhclust after loading a dataset.

The main clustering results are displayed in the heatmap with dendrogram tab, which is based on the `ComplexHeatmap` (Gu et al., 2016) package. By default, all numeric features will be displayed, with the features not used in the clustering shown directly below the main heatmap. In the sidebar, users can also select categorical features to be added as annotations in the top rows of the heatmap, as well as numeric features that contain missing values and that were identified when the dataset was loaded. As the heatmap shows the standardized data, users can investigate the characteristics of the clusters through a tab that contains plots showing the distributions of the original values of features across the clusters. The same tab also contains textual summaries of median, lower and upper quartiles for the features, and plots showing the distribution of data points for categorical features across clusters. Finally, as part of the main features of the clustering workflow, various internal validation metrics can be computed through the `clusterCrit` (Desgraupes, 2018) package. After these visualization and evaluation steps, users can quickly change the settings in the sidebar to compute another set of clusters and continue with the loop until satisfied with a solution. The original data annotated with the cluster labels can be downloaded at any point from the sidebar.

Besides these views, the tool also contains a tab for the computation of the Gap statistic (Tibshirani et al., 2001) as an additional evaluation metric and a tab that enables comparing differences across clusters for the features that were not used in the clustering (Dinno, 2017). Finally, basic support for data diagnostics is provided through tabs showing correlation heatmaps, PCA plots and projection of distance matrix, as well as a tabular view of clusters

for inspection and removal of data points. The interface contains multiple help points for each tab and also includes downloadable data examples (e.g. if the app is deployed in an internal server for multiple users), and the package documentation includes a visual tutorial and two examples of workflows using the exported functions.

Requirements and limitations

The version of the web app described here has the following requirements about data and limitations:

- **Handling missing data:** visxhclust was not designed to be used as a tool for a complete cluster-based workflow, thus the expectation is that any dataset loaded into the tool will have already been cleaned and pre-processed. However, the app is able to handle some deviations, such as mixed numeric and text values or missing rows, by enabling users to include features with such cases as annotations, but not allowing them to be selected for clustering.
- **Dataset size:** the web app also has, by design, limitations on the size of datasets.The aim was to provide interpretable outputs and support fast iterations, meaning that datasets starting with more than a few thousands of rows will likely result in larger computation times and messy outputs; the same applies to features – the expectation is that users will do the appropriate reduction of dimensionality for such cases before loading the data into the tool.
- **Guidance:** despite the inclusion of multiple help boxes, with external references to some of the packages mentioned above, the app is not an educational tool, nor does it include a step-by-step guidance on clustering. As such, it is possible that users will misinterpret results or settle on solutions that are distorted by the data.

Acknowledgements

This work was supported by the Health Data Research UK (grant ref: LOND1). We thank David Watson for helpful suggestions for the app and participants of the WHRI CompBio Code Review club for feedback on the package.

References

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., & Borges, B. (2021). *Shiny: Web application framework for R*. https://CRAN.R-project.org/package=shiny

Desgraupes, B. (2018). *clusterCrit: Clustering indices*. https://CRAN.R-project.org/package=clusterCrit

Dinno, A. (2017). *Dunn.test: Dunn's test of multiple comparisons using rank sums*. https://CRAN.R-project.org/package=dunn.test

Galili, T. (2015). Dendextend: An R package for visualizing, adjusting, and comparing trees of hierarchical clustering. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btv428

Gu, Z., Eils, R., & Schlesner, M. (2016). Complex heatmaps reveal patterns and correlations in multidimensional genomic data. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btw313

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). Visual analytics: Definition, process, and challenges. In *Lecture notes in computer science* (pp. 154–175). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-70956-5_7

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, *2*, 49–55.

Metsalu, T., & Vilo, J. (2015). ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap. *Nucleic Acids Research*, *43*(W1), W566–W570. https://doi.org/10.1093/nar/gkv468

Nijs, V. (2021). *Radiant: Business analytics using R and shiny*. https://cran.r-project.org/web/packages/radiant/index.html

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411–423. https://doi.org/10.1111/1467-9868.00293