SurvivalVolume: interactive volume threshold survival graphs

Matthew J. Wakefield[1,2]

[1]The Walter and Eliza Hall Institute
[2]The University of Melbourne

31 Oct 2016

Paper DOI: http://dx.doi.org/10.21105/joss.00111
Software Repository: https://github.com/genomematt/survivalvolume
Software Archive: http://dx.doi.org/10.5281/zenodo.200268

Summary

Treatment studies of cancer frequently use tumour volume to measure response to therapy. Therapeutic response will be apparent at different time points during the experiment. Progressive disease (increasing volume), stable disease and regression (reduction in volume) under therapy are important measures of response in addition to the overall time to reach a defined maximum volume. Traditional methods of presenting this data involve 3 unconnected graphs: line graphs of each individual, average volume of each group with standard error of the mean, and a Kaplan-Meier graph of time to maximum volume. Survival volume is a python package to produce an integrated plot of these three representations of the same data, and to provide interaction with the plots of volume to enhance exploration of outliers and subgroups that are of interest clinically.

Survival volume is written for python3 and uses matplotlib (Hunter 2007) and lifelines (C. 2016) Kaplan-Meier (Kaplan and Meier 1958) implementation for generating plots, mpld3 (Vanderplas 2016) for interactivity. Utility functions are provided for importing data from spreadsheets using xlrd (Machin 2016), preprocessing to provide consistent time scales for comparison of treatments, and conversion from volume measurements to survival format. Statistics are calculated using lifelines and scipy.stats (Jones et al. 2001–2001--). Pandas (McKinney 2010) data frames are used to provide flexible manipulation of data and use within Jupyter (Pérez and Granger 2007) notebooks is supported and encouraged.

Plot elements are presented in a visual hierarchy giving greatest weight to the information rich mean, and transparency is used to legibly overplot the confidence interval of the mean and the complete dataset. Interactivity through mouseover and plot zooming provides rich access to the full data set. By co-plotting the Kaplan-Meier representation with a shared x-axis endpoint and censoring events can be related between the plots enriching the information accessible about each event.

Survival volume is released under the GPLv3 and is available from GitHub and PyPI.

References

C., Davidson-Pilon. 2016. "Lifelines." *GitHub Repository*. https://github.com/camdavidsonpilon/lifelines; GitHub.

Hunter, J. D. 2007. "Matplotlib: A 2D Graphics Environment." *Computing In Science & Engineering* 9 (3).

IEEE COMPUTER SOC: 90–95.

Jones, Eric, Travis Oliphant, Pearu Peterson, and others. 2001–2001--. “SciPy: Open Source Scientific Tools for Python.” http://www.scipy.org/.

Kaplan, E. L., and Paul Meier. 1958. “Nonparametric Estimation from Incomplete Observations.” *Journal of the American Statistical Association* 53 (282): 457–81. doi:10.1080/01621459.1958.10501452.

Machin, John. 2016. “xlrd: A Library to Extract Data from Microsoft Excel (Tm) Files.” *GitHub Repository*. GitHub. https://github.com/python-excel/xlrd.

McKinney, Wes. 2010. “Data Structures for Statistical Computing in Python.” In *Proceedings of the 9th Python in Science Conference*, edited by Stéfan van der Walt and Jarrod Millman, 51–56.

Pérez, Fernando, and Brian E. Granger. 2007. “IPython: A System for Interactive Scientific Computing.” *Computing in Science and Engineering* 9 (3). IEEE Computer Society: 21–29. doi:10.1109/MCSE.2007.53.

Vanderplas, Jake. 2016. “mpld3: A D3 Viewer for Matplotlib.” *GitHub Repository*. GitHub. http://www.scipy.org/.