Assocplots: a Python package for static and interactive visualization of multiple-group GWAS results

Ekaterina A. Khramtsova[1,2] and Barbara E. Stranger[1,2]

Department of Medicine, Section of Genetic Medicine and Institute for Genomics and Systems Biology, The University of Chicago

Background

Over the last decade, genome-wide association studies (GWAS) have generated vast amounts of analysis results, requiring development of novel tools for data visualization. Quantile-quantile plots and Manhattan plots are classical tools which have been utilized to present GWAS results and identify variants significantly associated with traits of interest. However, static visualizations are limiting in the information that can be shown. Recently, dynamic, interactive visualization has become more widely adopted, however it has not yet become a routine part of GWAS data analysis. Interactive data visualization not only allows clearer representation of multidimensional data, but also encourages viewer's engagement from simple data browsing to providing a platform for answering specific scientific questions, in ways that static data cannot. Here we present a package for viewing GWAS results not only using classic static Manhattan and quantile-quantile plots, but also through interactive extension which allows to visualize data interactively. Our tool makes it possible to browse multiple charts in real-time to better understand the relationships among multiple groups.

Availability: The assocplots package is open source and distributed under the MIT license via GitHub along with examples, documentation and installation instructions: **https://github.com/khramts/assocplots**

Contact: eakhram@uchicago.edu

Static module features

Classic Manhattan plot

1. X-axis: chromosome and base pair, both numeric and alphabetical names (e.g. 1, 2, chr1, X), (Fig. 1)

2. Y-axis: Although -log10(p-value) is the most commonly used value for the y-axis, other values such as the effect size can be specified

3. Inverted Manhattan plot (also known as Chicago plot) for more convenient visualization of peak differences in two groups (Fig. 2).

Classic quantile-quantile plot

1. Multiple groups can be visualized on the same QQ plot for easier comparison (Fig. 3).

2. Genomic Inflation Factor, λ_{GC}, calculation: In GWAS population substructure and cryptic relatedness among subjects can lead to spurious errors, and genomic control method is commonly used correct the underlying population stratification (Devlin and Roeder, 1999).

3. The package allows to plot confidence intervals (CI) for either the null distribution or the experimental data. When multiple groups are plotted, CI can be displayed for each group.

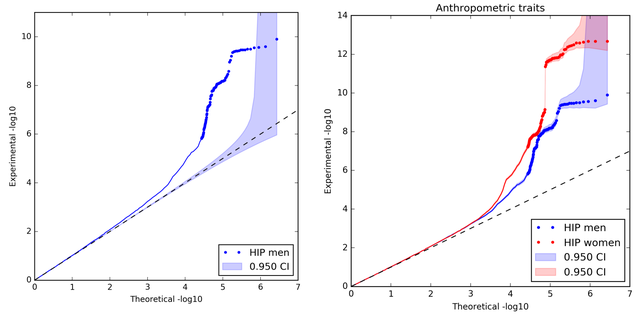

Figure 3. The package can be used to create classic QQ plots where the confidence interval is plotted for the null distribution, as well as for the experimental group, when multiple groups are plotted on the same figure.

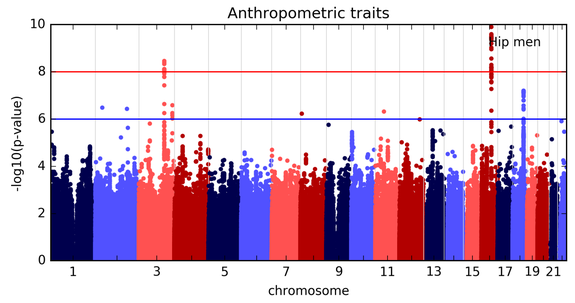

Figure 1. Example of a classic Manhattan plot, using data from the GIANT consortium (Randall JC, et al. 2013).

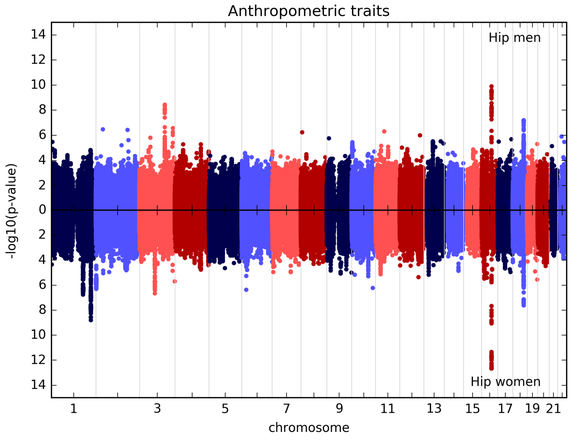

Figure 2. Example of a Chicago plot for two groups.

References

Delvin B and Roeder K. Genomic Control for association studies. (1999) Biometrics, 55:997-1004.

Randall JC et al. (2013) Sex-stratified Genome-wide Association Studies Including 270,000 Individuals Show Sexual Dimorphism in Genetic Loci for Anthropometric Traits. PLoS Genet 9(6): e1003500.

Turner, SD. (2014) qqman: an R package for visualizing GWAS results using Q-Q and Manhattan plots. *bioRxiv,* doi: 10.1101/005165.

Implementation

Assocplots is implemented as a package for the Python programming language. Interactive visualization is implemented via a Python interactive visualization library, bokeh (http://bokeh.pydata.org/), that targets modern web browsers; and data wrangling is implemented with Numpy and Pandas scientific computing python libraries. The package is designed to be used both in Jupyter notebooks and in command line. Visualizing GWAS data in a web-based document (notebook), ensures data analysis reproducibility and makes it conveniently sharable with collaborators via online repositories such as GitHub.

Interactive module features (Manhattan and QQ plot):

1. **Info pop-up:** Hovering over a point reveals information about the SNP/gene (Fig. 4), such as the name (SNP rs number or gene name), chromosome, base pair location, and the statistic reported on the y-axis (-log10 (pvalue) or effect size).

2. **Group comparison:** Selecting a set of SNPs in one graph automatically highlights those same SNPs in the other graph (a different phenotype, population, etc.). Additionally, a table is generated listing all the selected SNPs and information about those SNPs including the position, and the test statistic across groups.

3. **Zoom-in and -out:** Plotting many points on the same graph makes it difficult to discern one point from another, as it may be in a peak or in the lower portion of the plot which often is densely packed. To overcome this issue, the plot can be zoomed in using the mouse scroller when the mouse pointer is placed on the Manhattan plot.

4. **Sharing:** Interactive plots can be saved as notebooks and self-contained html files that can be shared with colleagues via usual sharing platforms (GitHub, Dropbox, Google Drive, etc.) and opened in any modern web browsers on any operation system.

Figure 4. A static view of interactive Manhattan plots for two groups. Selecting a group of SNPs in one of the groups highlights the same SNPs in the other group, and hovering over a dot shows the information associated with the SNP: rs number, base pair position, and a p-value. For the interactive version of this figure, see QR code below for an example.

Demo: http://khramts.github.io/output.html

Limitations

Generally, interactive visualization made through web browsers are limited by the number of objects they can smoothly display. In the current example we have selected the top (most significant) 1,000 SNPs in each of the two groups, with matching SNPs in the opposite group. To address this limitation, the package can be extended to a web application with dynamic data loading from a database/server. Dynamic data loading would allow to load SNP data in real time for a specific region of interest as the user zooms-in. By making this an open source package that is accessible via GitHub, we invite members of the scientific community to contribute and enhance the package's capabilities.

Funding NIH 3P50MH094267-04S1 and 1R01MH101820-S1