Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Application Note

Genome Analysis

G-Graph: An interactive genomic graph viewer

Peter A. Andrews[1,*], Joan Alexander[1], and Michael Wigler[1]

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Effective and efficient exploration of numeric values as a function of genomic position requires specialized software that has not been made available until now.

Results: We present G-Graph, an interactive genomic scatter plot viewer. G-Graph overlays multiple data series in one graph using different colors and markers, displays gene and cytoband information, allows easy changes to the appearance of data series, implements stack-based undo functionality, and saves user-selected application views as image and pdf files. G-Graph delivers smooth scrolling and zooming even for datasets with many millions of values. G-Graph runs under Linux, Mac OSX and Windows using Cygwin. G-Graph's target user is a clinician or researcher examining many copy number datasets to identify potentially deleterious mutations.

Availability: https://github.com/docpaa/mumdex/ or http://mumdex.com/ggraph/

Contact: andrewsp@cshl.edu (or paa@drpa.us)

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

G-Graph is an interactive application for plotting numeric data as a function of genomic position. The primary purpose for G-Graph is to enable efficient exploratory analysis of copy number and other datasets by clinicians and researchers. This requires the ability to quickly zoom in to and out of regions of interest, to display appropriate markers to distinguish samples and data types (i.e. points for binned copy number measurements or lines for segmented values), to change the stacking order of samples and to interpret events with an integrated display of gene annotation.

2 Methods

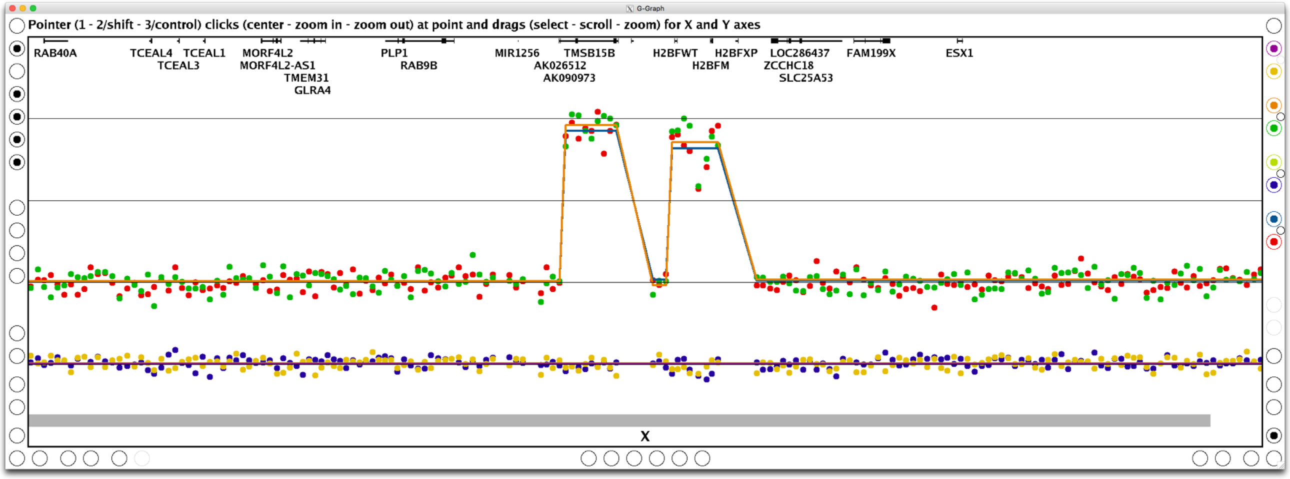
G-Graph is distributed as part of the MUMdex genome analysis software package (Andrews *et al.,* 2016). G-Graph compiles with no errors or warnings and functions properly in Linux or Unix with X11, Mac OSX with XQuartz, and Windows with Cygwin.

The only system requirements for G-Graph compilation are a C++11 (or later) compiler and a working X11 development environment. Python availability is optional, to check the code for potential problems (linting) during compilation. The ImageMagick convert application is optionally used to create a pdf file of all user selected views as well as individual png images, in addition to the xpm format image files that G-Graph natively outputs. G-Graph is not designed to produce vector graphics, but the output is nearly of publication quality. The Firefox web browser is launched when the user selects gene links to the UCSC Genome Browser (Kent *et al.,* 2002) in Linux and Windows, while Safari is used in Mac OSX.

At the core of G-Graph is a custom-built generic graphing module which is designed to be extensible via callback functions written by an application developer. G-Graph uses this extensibility to incorporate capabilities that are appropriate for genomic analysis. Advanced users with specific requirements can similarly change G-Graph functionality.

G-Graph is written in the C++ programming language to permit development at a high-level of abstraction without sacrificing run-time efficiency. G-Graph code wraps low-level details such as bytes, memory locations, fundamental types and library interfaces in higher-level objects representing concepts such as fonts, windows, datasets, graphs, etc.

G-Graph employs the low-level and stable X11 Xlib for visualization without the aid of a widget toolkit such as Tk. This choice increases G-Graph portability and ease of installation while allowing unique modes of interaction which give G-Graph a distinctive feel that is unchanged under different window managers. G-Graph suffers minimal degradation of responsiveness even for remote display over slow networks because X11 enables very efficient data display.



G-Graph

Pointer (1 - 2/shift - 3/control) clicks (center - zoom in - zoom out) at point and drags (select - scroll - zoom) for X and Y axes

Figure 1: A zoomed in G-Graph application view showing inherited X chromosome amplification only in family females (see text for a full description).

3 Results

Figure 1 is a G-Graph screenshot showing a megabase portion of the X chromosome from a 500,000 bin copy number analysis of one family from the autism Simons Simplex Collection (Fischbach *et al*., 2010), processed with MUMdex alignment and copy number software. The four horizontal lines in the figure indicate copy number values of 1, 2, 3 and 4. The father (blue markers) and son (yellow) have a copy number value of 1, while the mother (red) and daughter (green) are mostly at a copy number value of 2, but there are also amplifications in the daughter that were inherited from the mother. The segmentation is rendered as differently colored lines.

The visible G-Graph application interface consists of a central graph region within the black rectangle, a status line at the top which displays application messages and a number of grouped radio button controls along the remaining three borders. By default, the status line displays a tooltip message for each interface element that the user's pointer hovers over. The controls and status display disappear when the pointer focus exits the window, to provide an uncluttered view of application content. The window can be adjusted to any size and aspect ratio possible on the user's screen.

The radio buttons are grouped by function and come in togglable (turn on and off) and non-togglable (perform an action) varieties. The top right control implements an undo function that reverts to previously displayed views. Below that, colored controls individually select which data series are displayed and change series stacking order. The bottom right controls change the appearance of the markers and lines. The bottom left controls perform zoom-outs to the full data range and toggle the display of grid lines, axis labels and a logarithmic scale y axis. The top left controls save the current view and toggle tooltip, coordinate, ratio line, chromosome label, gene, and cytoband display. The central group of controls along each axis perform discrete jumps and continuous scrolling. Some controls are inactive and shaded in a lighter color when inapplicable, such as a zoom out control for a fully zoomed out view.

The G-Graph interface also includes every pixel of the application window that is not part of a radio control, and is used to center, scroll and zoom the view. For instance, a primary button pointer drag in the graphing region will zoom in to the selected region while a drag with the secondary button (or shift) depressed will scroll the view for both axes, and a drag with the tertiary button (or control) depressed produces continuous zooms. Similar actions along the borders affect only the closest axis. A primary button pointer click along the top or bottom border of the application will center the x axis at the selected point, while one with the secondary / tertiary button depressed will zoom in / out only the x axis. Clicks along y axis borders behave similarly and clicks in the graphing region affect the ranges of both axes. The status line shown in Figure 1 succinctly describes this pointer action behavior (for the graph region), which was designed for rapid multiscale exploration.

The genic structure and gene names are displayed at the top of the graph region when the view is sufficiently zoomed in. If the user's pointer hovers over any gene name, the standard description of the gene is displayed in the status line. If a gene name is clicked with the pointer, the user's web browser is instructed to load the UCSC gene browser view for the gene. Cytobands are displayed as colored thick horizontal bands at the bottom of the graph region (extending almost to right edge of graph in the figure).

To demonstrate G-Graph usage, the command line for Figure 1 was:

```
ggraph cn hg19.fa abspos,ratio,seg {m,f,d,s}.txt
```

where cn indicates that copy number ratio lines and genome information should be shown (plain and genome are alternatives) and hg19.fa is the reference genome file used. Next, the x axis absolute genome position (abspos - ignoring all chromosomes less than 40Mb in size by default) and two y axis dependent variable identifiers (ratio, seg) are given. Numeric column identifiers can alternatively be used if there is no header line in the data files. Placing a :p or an :l after any y axis column name explicitly selects point or line display, respectively. When exactly two y axis column names are specified without qualification, the second is by default displayed as lines instead of points. The input text data files are m.txt, f.txt, d.txt and s.txt (mother, father, daughter, son). As can be deduced from the usage, one or more y axis values can be shown from one or more arbitrarily-named data files. The files must be whitespace-separated and tabular in form, with numeric x values in one column and corresponding y values in other columns, in any order, with unreferenced columns permitted.

The dataset used here and UCSC gene and cytoband definition files for hg19 and hg38 are available as supplementary materials. A tutorial at http://mumdex.com/ggraph/ describes G-Graph installation and use.

Funding

This work was supported by grants to M. Wigler from the Breast Cancer Research Foundation and from the Simons Foundation Autism Research Initiative (SF235988).

Conflict of Interest: none declared.

References

Andrews PA, Iossifov I, Kendall J, Marks S, Muthuswamy L, Wang Z, Levy D, Wigler M. (2016) MUMdex: MUM-based structural variation detection. *bioRxiv* 078261; doi: http://dx.doi.org/10.1101/078261

Fischbach GD, Lord C. (2010) The Simons Simplex Collection: a resource for identification of autism genetic risk factors. *Neuron*, **68** (2), 192-195

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. (2002) The human genome browser at UCSC. *Genome Res.* **12** (6), 996-1006.