

# RIGHT: an *HTML* canvas and *JavaScript*-based interactive data visualization package for linked graphics

ChungHa Sung<sup>1</sup>, TaeJoon Song<sup>2</sup>, Jae W. Lee<sup>1</sup>, and Junghoon Lee<sup>3\*</sup>

1. Sungkyunkwan University (SKKU), Suwon, Gyeonggi-Do, 440-746, South Korea

2. Samsung Electronics, Hwaseong-Si, Gyeonggi-Do, 143-130, South Korea

3. Merck Research Laboratories, Rahway, NJ, 08901, U.S.A.

\*Contact author: [jung\\_hoon\\_lee@merck.com](mailto:jung_hoon_lee@merck.com)

**Keywords:** interactive data visualization, linked graphics, *HTML* canvas, *JavaScript*

Interactive data visualization has received broad interest in the *R* community due to its obvious benefits over static visualization: more information can be delivered concisely and intuitively by user engagement. As a result, various *R* packages supporting single-layer, multi-layer, and linked graphics have been developed, including **rCharts**, **iPlots**, **cranvas**, **ggvis**, **animint** and **googleVis** [1]. **R** Interactive Graphics via HTml (**RIGHT**, <https://code.google.com/p/r-interactive-graphics-via-html/>) is an interactive data visualization package for linked graphics based on *HTML* canvas and *JavaScript*. It provides an *R* API similar to base graphics to easily construct various interactive plots, including scatter, line, bar, pie, and box-whisker plots.

This poster presents an overview of **RIGHT** and the *JavaScript* data structure that enables linked graphics. **RIGHT** is the first package that implements linked graphs using *HTML* canvas and *JavaScript*. Linked graphics help answer obvious questions a collection of plots tend to raise: how one point in the plot is related to another point in another plot. *HTML* canvas and *JavaScript* make it possible to deliver the visualization to various platforms, including mobile devices, since they are standard web technologies supported by most modern web browsers (albeit some remaining compatibility issues). This approach can also benefit from the improvement of *JavaScript* performance every generation, driven by various web applications with ever increasing complexity and sophistication.

## References

[1] Toby Dylan Hocking, <https://github.com/tdhock/interactive-tutorial> (as of March 16, 2014).