

Configurable point rasterization for large scatterplots



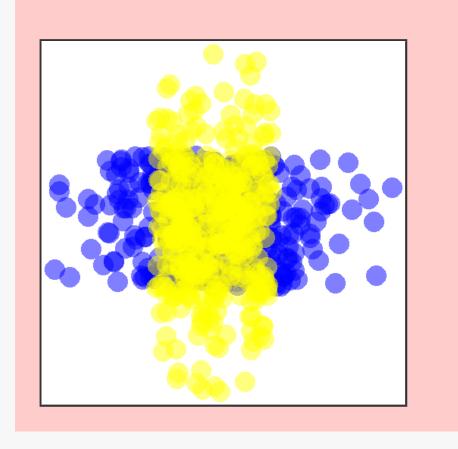
Tereza Kulichová

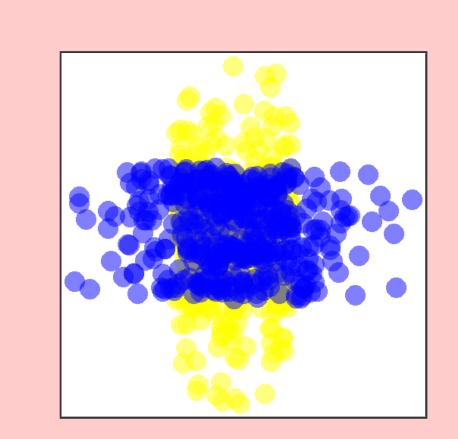
BACKGROUND

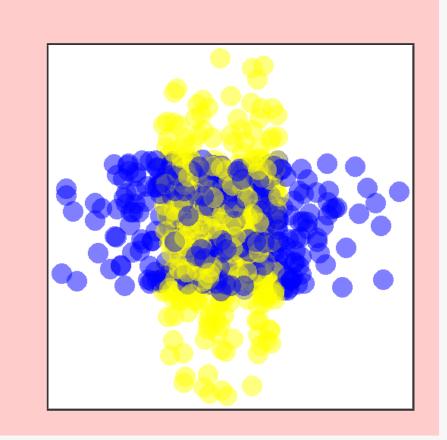
Until recently, scatterplots could only be plotted in R using its very slow base plot function.

The **speed** problem was solved by package scattermore. However, scattermore had only a single-purpose **API** and it didn't prevent **overplotting** – overlapping of the data in the visualization, making it difficult to see individual data.

The problem of overplotting, its symmetric case, and its primitive insufficient solution by randomizing point order are visible below.





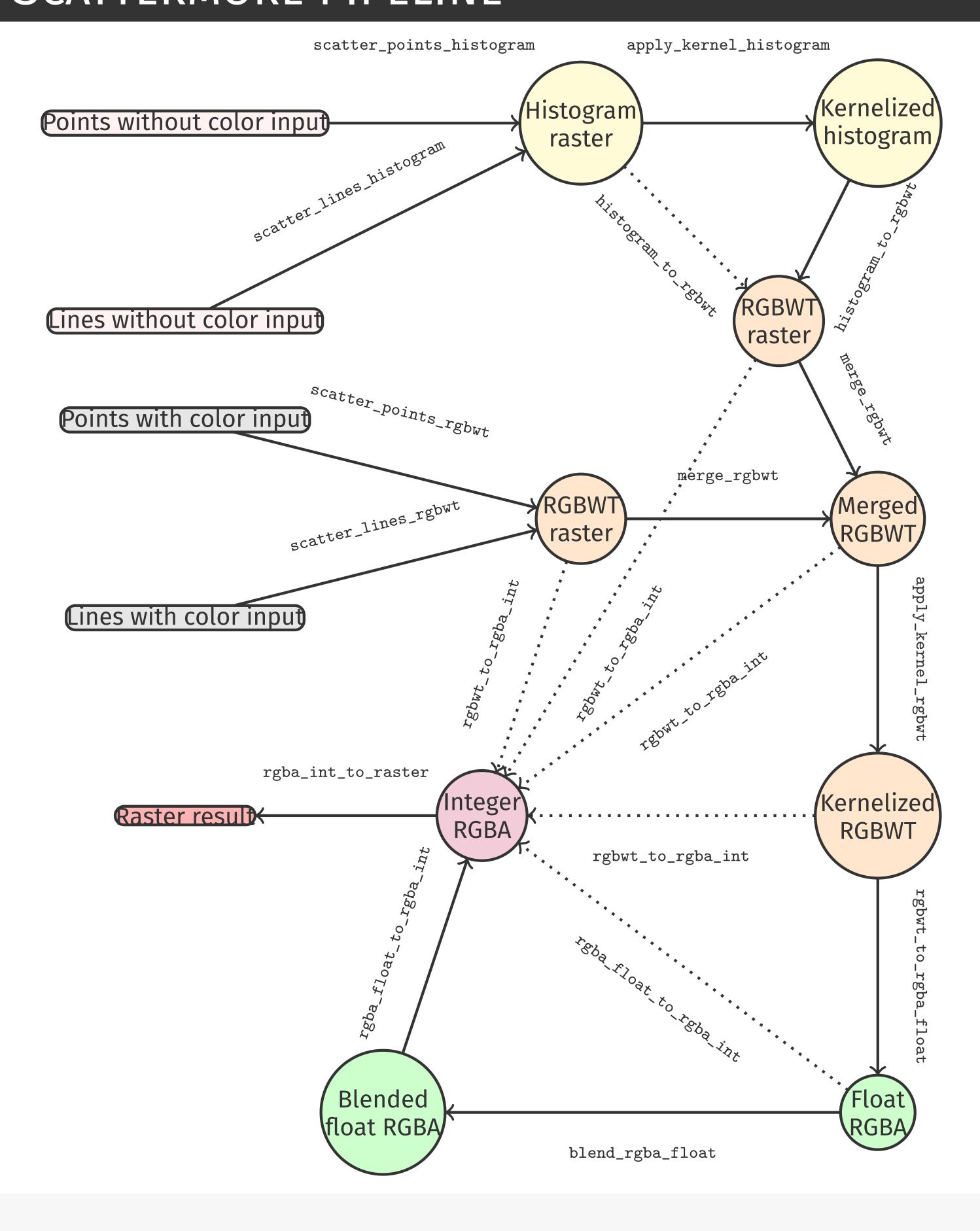


THESIS GOALS

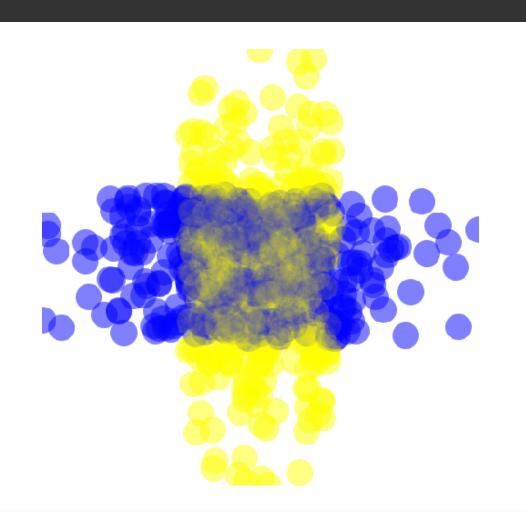
We redesign scattermore with the following requirements.

- **High-customizable API.** We assign each subtask to a single method to ensure more variation in output for the user.
- Reduction of overplotting. We introduce the new RGBWT data format, which combines images equally.
- Fast plotting. We implement the computationally expensive operation in C language.

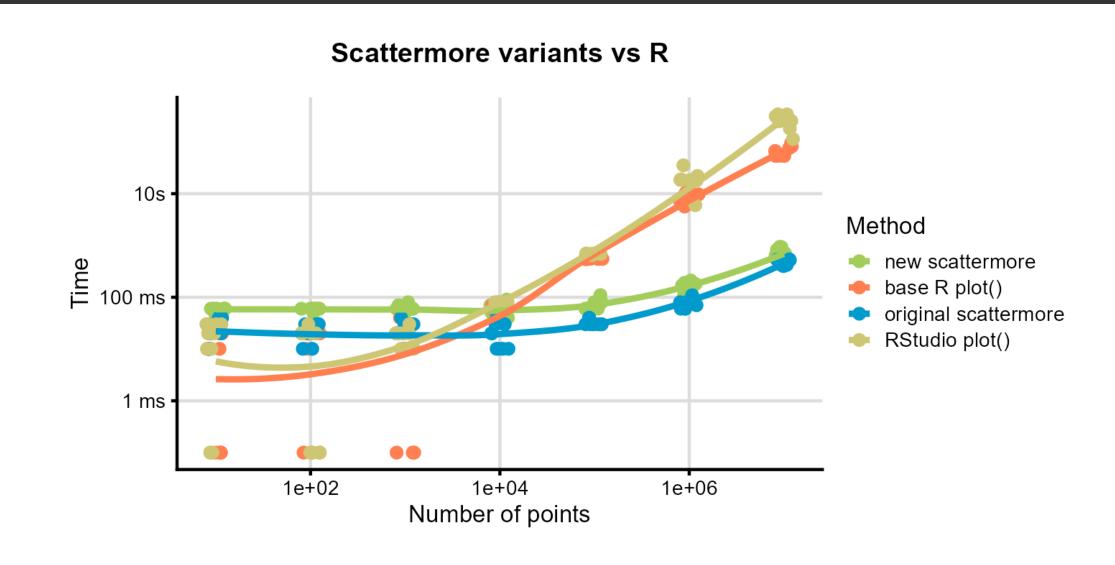
SCATTERMORE PIPELINE



RESULT: SOLUTION TO OVERPLOTTING AND BETT



RESULT: IMPROVED PERFORMANCE



Comparison of the speed difference when plotting points among R, RStudio, original, and new scattermore's version.

MAIN RESULT

Scattermore offers a highly customizable API, solves the problem of overplotting, and is much faster than base R functions.

CONCLUSIONS

- Scattermore enables users to blur images, blend or merge them according to their preferences and, at the same time, sufficiently fast.
- It is especially useful in the cytometric community for interactive analysis of large biological data.
- Scattermore is available on GitHub as a development version and as R's official CRAN package version.

Possible future work:

- Working directly with 32-bit float data, avoiding the frequent conversions to 64-bit floats used in R.
- Porting some operations to GPU.
- Possibility to plot Bezier curves and filled polygons.

