Package 'scattermore'

October 14, 2022

Title Scatterplots with More Points

Version 0.8

Description C-based conversion of large scatterplot data to rasters. Speeds up
plotting of data with millions of points.

URL <https://github.com/exaexa/scattermore>

Imports ggplot2, scales, grid, grDevices, graphics

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation yes

Author Mirek Kratochvil [aut, cre] (<<https://orcid.org/0000-0001-7356-4075>>)

Maintainer Mirek Kratochvil <exa.exa@gmail.com>

Repository CRAN

Date/Publication 2022-02-14 10:20:02 UTC

R topics documented:

1

---

geom_scattermore   *geom_scattermore*

Description

[ggplot2::ggplot()](#) integration. This cooperates with the rest of ggplot (so you can use it to e.g. add rasterized scatterplots to vector output in order to reduce PDF size). Note that the ggplot processing overhead still dominates the plotting time. Use [geom_scattermost()](#) to tradeoff some niceness and circumvent ggplot logic to gain speed.

Usage

```
geom_scattermore(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  interpolate = FALSE,
  pointsize = 0,
  pixels = c(512, 512)
)
```

Arguments

mapping, data, stat, position, inherit.aes, show.legend, ...	
	passed to [ggplot2::layer()](#)
na.rm	Remove NA values, just as with [ggplot2::geom_point()](#).
interpolate	Default FALSE, passed to [grid::rasterGrob()](#).
pointsize	Radius of rasterized point. Use `0` for single pixels (fastest).
pixels	Vector with X and Y resolution of the raster, default `c(512,512)`.

Details

Accepts aesthetics `x`, `y`, `colour` and `alpha`. Point size is fixed for all points. Due to rasterization properties it is often beneficial to try non-integer point sizes, e.g. `3.2` looks much better than `3`.

Examples

```
library(ggplot2)
library(scattermore)
ggplot(data.frame(x=rnorm(1e6), y=rexp(1e6))) +
  geom_scattermore(aes(x,y,color=x),
                   pointsize=3,
```

```
                        alpha=0.1,
                        pixels=c(1000,1000),
                        interpolate=TRUE) +
    scale_color_viridis_c()
```

---

geom_scattermost   *geom_scattermost*

Description

Totally non-ggplotish version of [`geom_scattermore()`](#), but faster. It avoids most of the ggplot processing by bypassing the largest portion of data around any ggplot functionality, leaving only enough data to set up axes and limits correctly. If you need to break speed records, use this.

Usage

```
geom_scattermost(
  xy,
  color = "black",
  interpolate = FALSE,
  pointsize = 0,
  pixels = c(512, 512)
)
```

Arguments

xy	2-column object with data, as in [`scattermore()`](#).
color	Color vector (or a single color).
interpolate	Default FALSE, passed to [`grid::rasterGrob()`](#).
pointsize	Radius of rasterized point. Use `0` for single pixels (fastest).
pixels	Vector with X and Y resolution of the raster, default `c(512,512)`.

Examples

```
library(ggplot2)
library(scattermore)
d <- data.frame(x=rnorm(1000000), y=rnorm(1000000))
x_rng <- range(d$x)
ggplot() +
  geom_scattermost(cbind(d$x,d$y),
                    color=heat.colors(100, alpha=.01)
                          [1+99*(d$x-x_rng[1])/diff(x_rng)],
                    pointsize=2.5,
                    pixels=c(1000,1000),
                    interpolate=TRUE)
```

scattermore	*scattermore*

Description

Convert points to raster scatterplot rather quickly.

Usage

```
scattermore(
  xy,
  size = c(512, 512),
  xlim = c(min(xy[, 1]), max(xy[, 1])),
  ylim = c(min(xy[, 2]), max(xy[, 2])),
  rgba = c(0L, 0L, 0L, 255L),
  cex = 0,
  output.raster = TRUE
)
```

Arguments

xy	2-column float matrix with point coordinates. As usual with rasters in R, X axis grows right, and Y axis grows DOWN. Flipping `ylim` causes the usual mathematical behavior.
size	2-element vector integer size of the result raster, defaults to `c(512,512)`.
xlim, ylim	Float limits as usual (position of the first pixel on the left/top, and the last pixel on the right/bottom). You can easily flip the top/bottom to the "usual" mathematical system by flipping the `ylim` vector.
rgba	4-row matrix with color values of 0-255, or just a single 4-item vector for `c(r,g,b,a)`. Best created with `col2rgb(..., alpha=TRUE)`.
cex	Additional point radius in pixels, 0=single-pixel dots (fastest)
output.raster	Output R-style raster (as.raster)? Default TRUE. Raw array output can be used much faster, e.g. for use with png::writePNG.

Value

Raster with the result.

Examples

```
library(scattermore)
plot(scattermore(cbind(rnorm(1e6),rnorm(1e6)), rgba=c(64,128,192,10)))
```

scattermoreplot *scattermoreplot*

Description

Convenience base-graphics-like layer around scattermore. Currently only works with linear axes!

Usage

```
scattermoreplot(
  x,
  y,
  xlim,
  ylim,
  size,
  col = grDevices::rgb(0, 0, 0, 1),
  cex = 0,
  pch = NULL,
  xlab,
  ylab,
  ...
)
```

Arguments

x, y, xlim, ylim, xlab, ylab, ...	used as in `graphics::plot()` or forwarded to `graphics::plot()`
size	forwarded to `scattermore()`, or auto-derived from device and plot size if missing (the estimate is not pixel-perfect on most devices, but gets pretty close)
col	point color(s)
cex	forwarded to `scattermore()`
pch	ignored (to improve compatibility with `graphics::plot()`

Examples

```
# plot an actual rainbow
library(scattermore)
d <- data.frame(s=qlogis(1:1e6/(1e6+1), 6, 0.5), t=rnorm(1e6, pi/2, 0.5))
scattermoreplot(
  d$s*cos(d$t),
  d$s*sin(d$t),
  col=rainbow(1e6, alpha=.05)[c((9e5+1):1e6, 1:9e5)],
  main="scattermore demo")
```

Index