

SysBioSig Theme

Karol Nienaltowski

Installation

ITRC package can be installed using `devtools` package as following

```
install.packages("devtools") # if devtools package is not installed
require(devtools)
devtools::install_github("stork119/SysBioSigTheme", auth_token = [AUTHORISATION_TOKEN])
```

[AUTHORISATION_TOKEN] can be obtained on request after sending an email on address k.nienaltowski@sysbiosig.org (mailto:k.nienaltowski@sysbiosig.org).

To attach `SysBioSigTheme` functions use:

```
library(SysBioSigTheme)
```

Normalise data

To read data tables generated by Cell Profiler and IPIQA it is necessary to use normalisation function

```
data <- normalize_data(read.table([PATH_TO_FILE],
                                header=TRUE, sep=",")$data
```

Themes

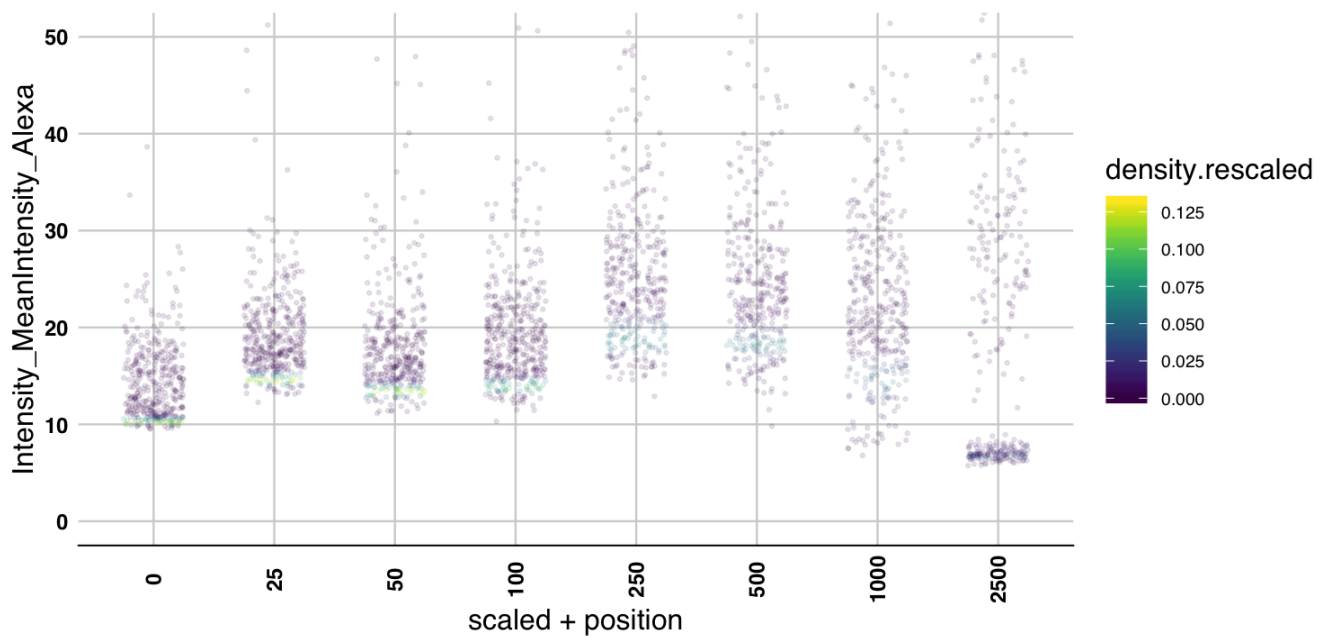
Instead of `theme_jetka()` you can use new theme function `theme_sysbiosig()`

Scatter plots

In order to plot novel scatter plots use functions `ScatterBoxplotGGplot` :

```
ScatterBoxplotGGplot(data = SysBioSigTheme::data.jarek,
                      x_ = "stimulation.1.1",
                      y_ = "Intensity_MeanIntensity_Alexa") +
  coord_cartesian(ylim = c(0,50))
```

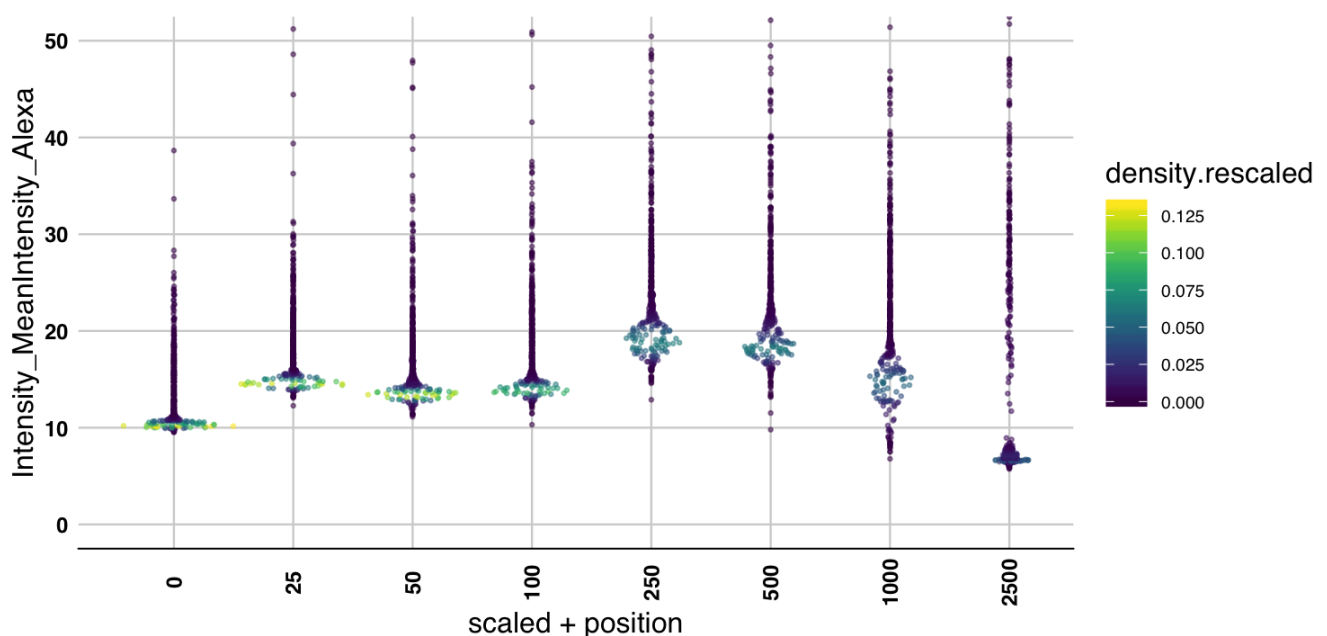
```
## Joining, by = "stimulation.1.1"
```



Or `ScatterViolinGGplot` :

```
ScatterViolinGGplot(data = SysBioSigTheme::data.jarek,
                     x_ = "stimulation.1.1",
                     y_ = "Intensity_MeanIntensity_Alexa") +
  coord_cartesian(ylim = c(0,50))
```

```
## Joining, by = "stimulation.1.1"
```

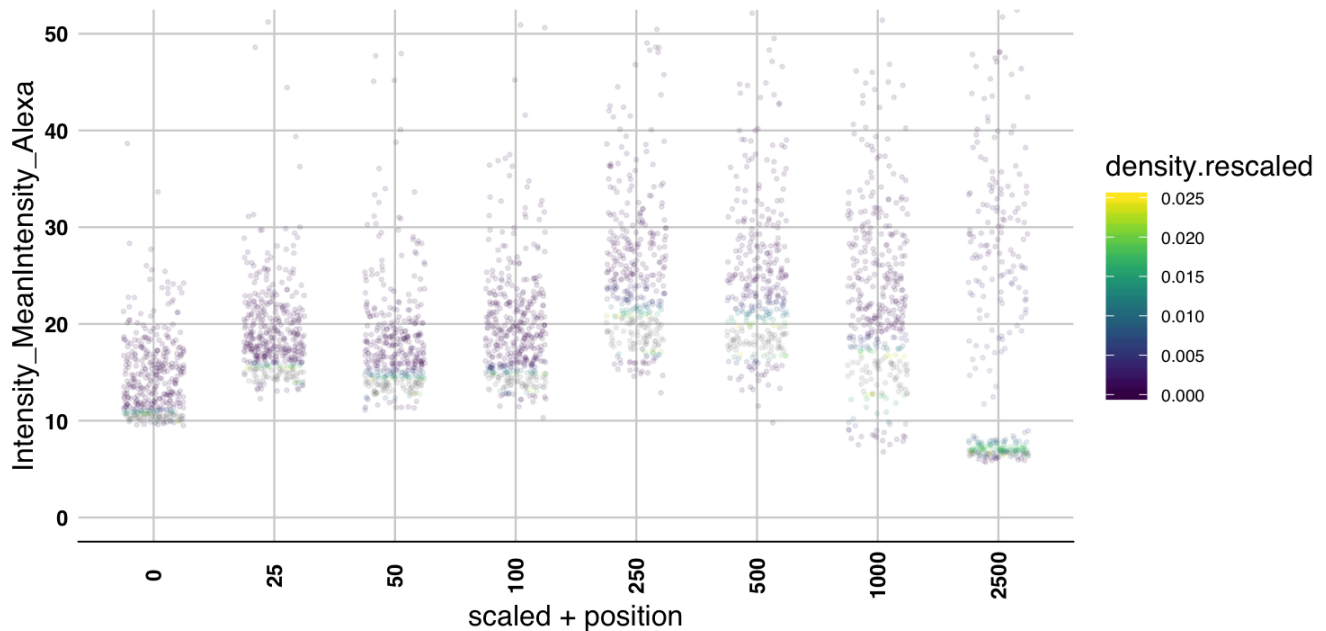


Additional Parameters

You can change limits of the densities colors using parameter `colors.limits` :

```
ScatterBoxplotGGplot(data = SysBioSigTheme::data.jarek,
                      x_ = "stimulation.1.1",
                      y_ = "Intensity_MeanIntensity_Alexa",
                      colors.limits = c(0, 0.025)) +
coord_cartesian(ylim = c(0,50))
```

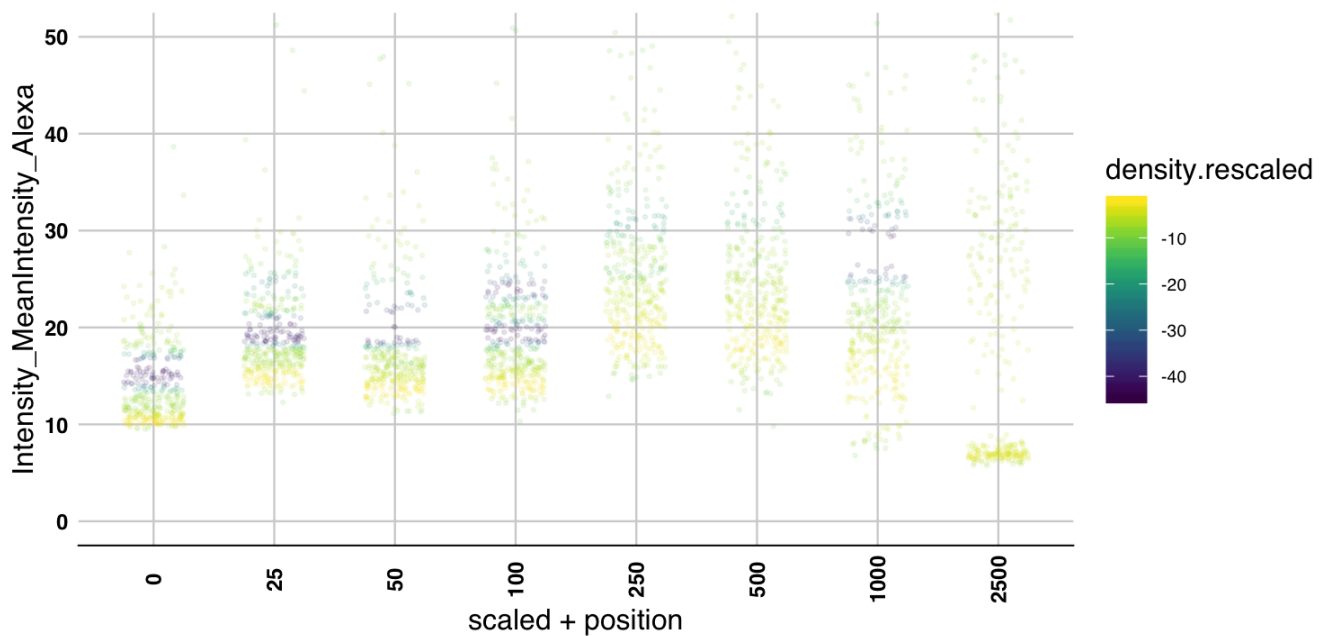
```
## Joining, by = "stimulation.1.1"
```



You can rescale colors palette densities colors using parameter `density.rescale.fun`. Parameter `density.rescale.fun` defines a function used for rescaling signals in plots. There are three built-in functions, that can be chosen: (1) 'numeric', (2) logarithmic - with base defined in `density.rescale.fun.args` - default: $e = \exp(1)$. Function must be defined as a lambda construct `function(x, ...){...}`. If you want to use logarithmic scale of densities colors set parameter `density.rescale.fun = "log"`.

```
ScatterBoxplotGGplot(data = SysBioSigTheme::data.jarek,
                      x_ = "stimulation.1.1",
                      y_ = "Intensity_MeanIntensity_Alexa",
                      density.rescale.fun = "log") +
coord_cartesian(ylim = c(0,50))
```

```
## Joining, by = "stimulation.1.1"
```

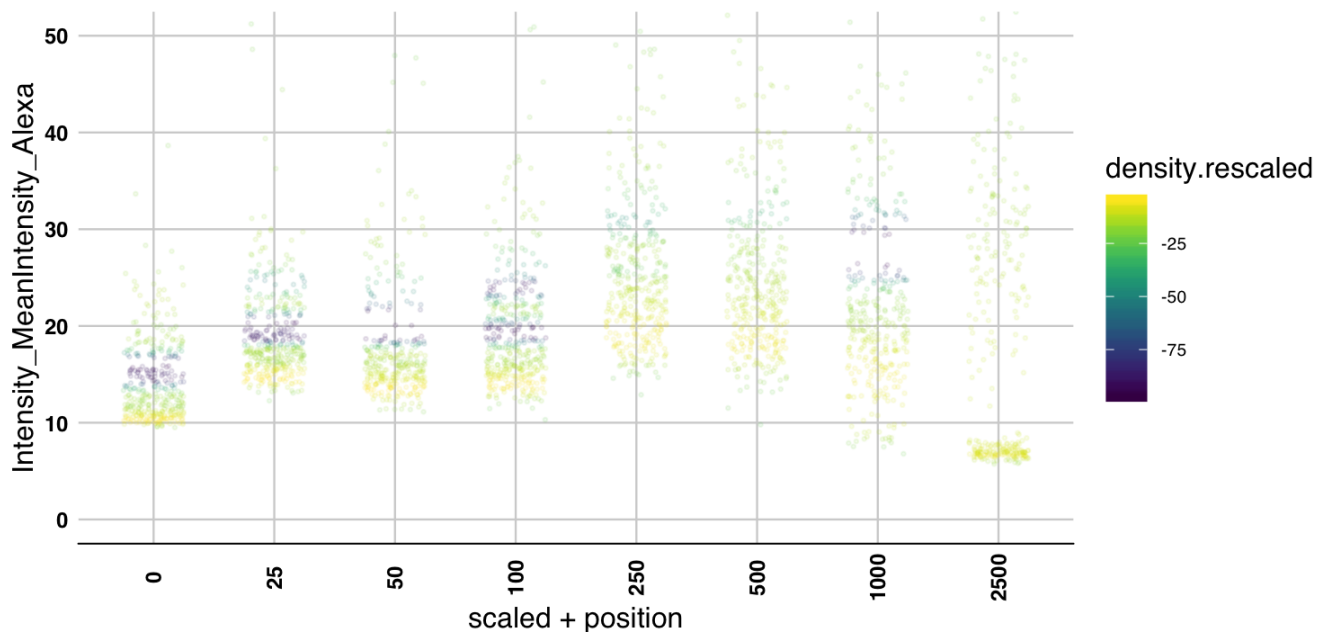


You can also use your own function. For example

```
density.rescale.fun = function(x){5*log(x, base = 10)}.
```

```
ScatterBoxplotGGplot(data = SysBioSigTheme::data.jarek,
  x_ = "stimulation.1.1",
  y_ = "Intensity_MeanIntensity_Alexa",
  density.rescale.fun = function(x){5*log(x, base = 10)}) +
  coord_cartesian(ylim = c(0,50))
```

```
## Joining, by = "stimulation.1.1"
```



You can modify themes by adding special three parameters. All possible parameters of theme function you can find in `theme` description `?theme`. For instance you can manipulate position of the legend using `legend.position`. You can choose ("none", "left", "right", "bottom", "top", or two-element numeric vector). For instance legend is removed, when `legend.position = "none"`.

```
ScatterBoxplotGGplot(data = SysBioSigTheme::data.jarek,  
  x_ = "stimulation.1.1",  
  y_ = "Intensity_MeanIntensity_Alexa",  
  legend.position = "none") +  
  coord_cartesian(ylim = c(0,50))
```

```
## Joining, by = "stimulation.1.1"
```

