Interactive Figures with R

Bingbing Yuan

Apr 5, 2018

Packages:

- ❖ rbokeh
- ❖ ggvis
- ❖ shiny

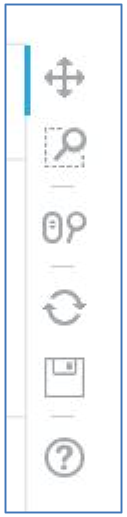
- Packages were chosen based on:
  1. Create interactive figures w/o requiring HTML, CSS, or JavaScript knowledge
  2. Input data types: text file from RNA-seq analysis

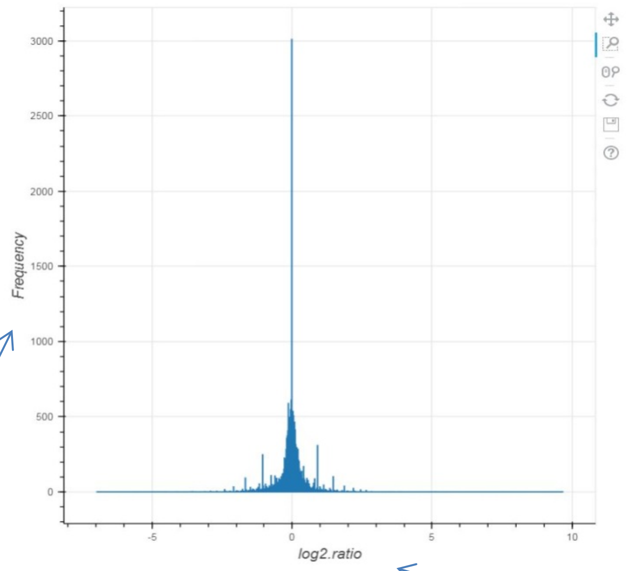
Install rbokeh with library(devtools) -> install_github("hafen/rbokeh")

rbokeh

- R interface to bokeh
  - htmlwidgets ,maps,ggplot2, scales, etc.
- Initiated with figure()
- Add layers (prefixed with ly_)
- Interaction: tools

```
# draw histogram w/ rbokeh
h <- figure ()  %>%
 ly_hist(log2.ratio, data=d, breaks=1000 )
h
```





Frequency

Log2.ratio

rbokeh

- figure(data = NULL, title = NULL, xlab = NULL, ylab = NULL, xlim = NULL, ylim = NULL, legend_location = "top_right", tools = c("pan", "wheel_zoom", "box_zoom", "reset", "save", "help"), ...)

  Tools:

    default: above

    additional tools: box_select, lasso_select

- layers (prefixed with ly_):
  - ly_points, ly_lines, ly_abline, ly_hist, ly_boxplot, etc.
    - ly_point (fig, x, y = NULL, data, color, alpha, size, url, hover, ...)

rbokeh Demo:

- Histogram with zoom in function:
  - rbokeh_hist.R
- Hover and access to gene information:
  - rbokeh_hover.R
- Draw volcano plot plus highlight/display DE genes
  - rbokeh.volcanoPlot.R
- Select subset genes with box/lasso selection
  - rbokeh_select.R

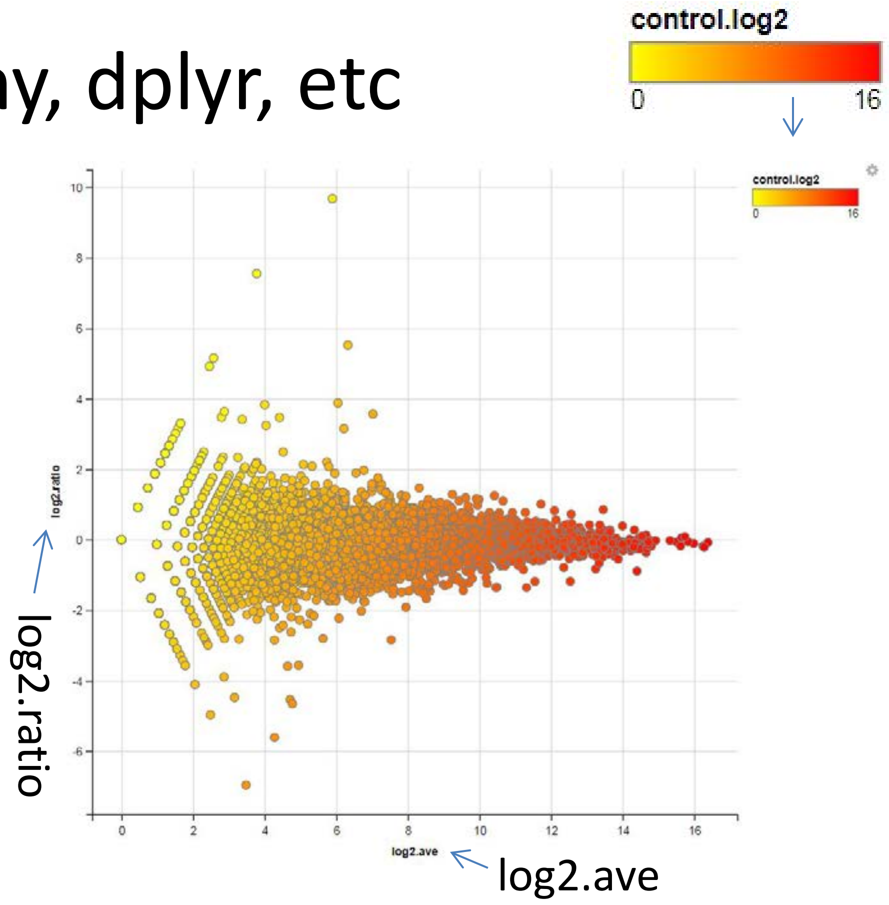
rbokeh
multiple plots

- grid_plot(figs.list)
- layout: nrow, ncol, byrow
  - grid_plot (figs, ncol=2, byrow=TRUE)
- link_data
  - TRUE: link all data sources that are common among the different figures in the plot
- Others:
  - grid_plot(figs, width = 600, height = 300, xlim = c(0,20), ylim = c(-10,10))

- Demo:
  - Compare two MAplots:
    - rbokeh_compare2conds_allgenes.R
    - Small number of genes: rbokeh_compare2conds.R
  - Compare volcano plot with MA plot
    - rbokeh_link_maplot_w_volcano_plots.R

ggvis

- Combine ggplot2, shiny, dplyr, etc
- Starts with ggvis()
- Draw plot with layer:
  - layer_points, layer_text
    - lay_points:
      - Stroke, fill, opacity
  - scales:
    - scale_numeric
    - scale_nominal



```
d %>%     ggvis (~log2.ave, ~log2.ratio, fill =~control.log2, stroke :="grey") %>%
          layer_points() %>%
          scale_numeric("fill", range = c("yellow", "red"))
```

Mapping = ~variable *Setting :=2 or :="red"*

ggvis interactivity

- Interactive widgets (prefixed with input_):
  - input_slider()
  - input_checkbox(), input_select(), input_radiobuttons()
  - input_text(), input_numeric()

ggvis Demo

- Based on user's input, change the size of points inside a scatter plot
  - With input_slider
    - ggvis_pointSizeSelector.R
- Based on user's inputs, change the color of the points in a scatterplot
  - With input_text
    - ggvis_text.R
  - With input_radiobutton
    - ggvis_radiobutton.R
- Update scatterplot with user selected samples
  - input_select:
    - ggvis_selectSamples.R

Shiny

- Open source R package
- Powerful package for create interactive plots
- Syntax more complex than rbokeh and ggvis

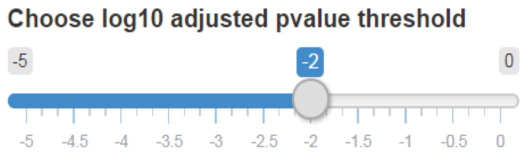
example.R

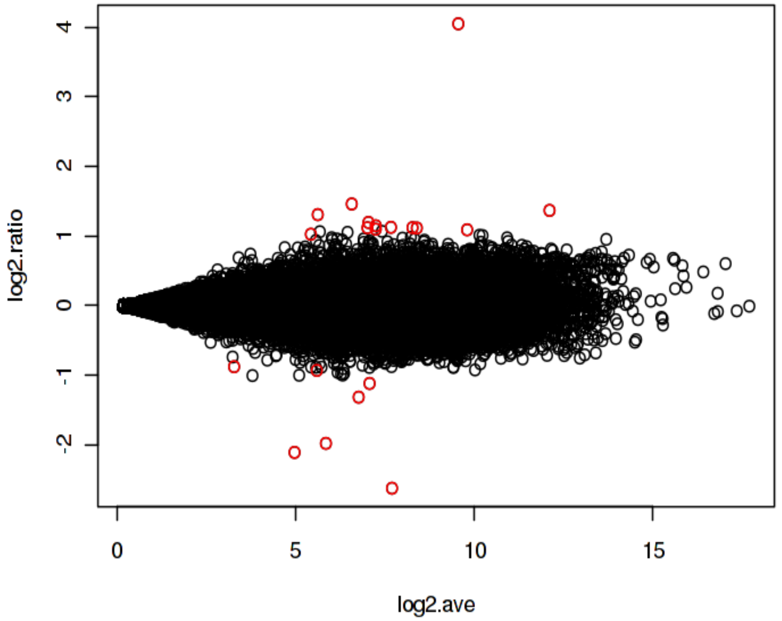
```
library(shiny)

ui <- fluidPage(
sliderInput(
  inputId = “usrpadj",
  label = "Choose log10 adjusted pvalue threshold",
  value = -2 , min = -5, max = 0, step=0.25),
 plotOutput(“maplot") )


server <- function(input, output) {
   output$maplot <- renderPlot({
   # scatter plot
   plot(log2.ave, log2.ratio)
   # select DE genes
   s = subset(d, d$padj <= 10^ input$usrpadj )
   # highlight DE genes
   points(s$log2.ave, s$log2.ratio, col="red")
 })

shinyApp(ui = ui, server = server)
```





UI input

- sliderInput
- checkboxGroupInput, checkboxInput
- radioButtons
- selectInput
- textAreaInput
- submitButton
- …

Outputs:
render*() and *Output functions work together to add R output to UI

- renderPlot <- plotOutput
- renderText <- textOutput
- renderDataTable <- dataTableOutput
- renderImage <- imageOutput
- …

Shiny Demo

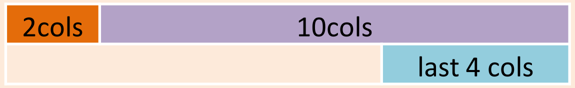
- Draw MAplot, DE genes are based on user input
  - With sliderInput, plotOutput, renderPlot
  - shiny_MAplotAdjustedByadj.R
- Upload a file, and display its content in table format enable sorting and selecting function:
  - With fileInput, dataTableOuput, renderDataTable
  - shiny_uploadFile.R

Shiny Demo

- Upload DESeq2 output file, select cutoff s(log2ratio, pvalue) -> draw plots (MAplot, heatmap, volcano plots), show selected genes, and download file.
  – shiny_processDESeq2outputFile.R

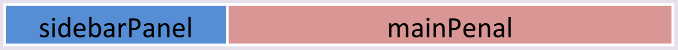
UI Layout

- Grid layout:
  - fluidRow()
  - column() : 12 columns in total

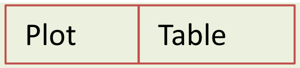


```
fluidPage(
        fluidRow(
                column(2, "2cols),
                column(10, "10cols" ) ),
        fluidRow(
                column(4, offset =8,  “last 4cols“ )))
```

- Simple Sidebar Layout:
  - sidebarLayout(
    sidebarPanel (),
    mainPenal ())

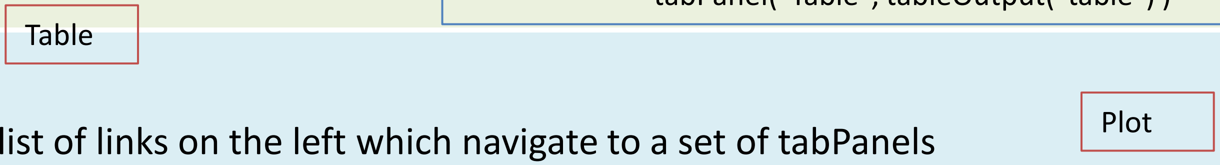


- Segmenting layout:
  - tabsetPanel ()
    - Under mainPanel

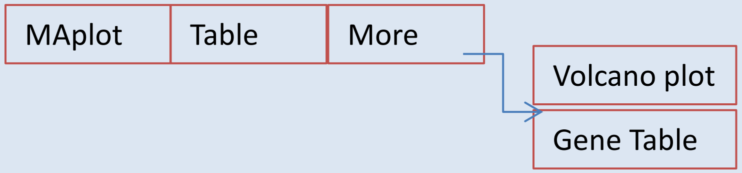


```
tabsetPanel(
        tabPanel("Plot", plotOutput("plot")),
        tabPanel("Table", tableOutput("table") )
```

  - navlistPanel()
    - provides a list of links on the left which navigate to a set of tabPanels displayed to the right



- multiple top-level components:
  - navbarPage



```
navbarPage(
        tabPanel(“MAplot“, plotOutput(“maplot”)),
        tabPanel(“Heatmap“, plotOutput(“heatmap”)),
        navbarMenu(“More",
                tabPanel(“Volcano plot“, plotOutput(“volcano”)),
                tabPanel(“Gene Table”, tableOutput(“genesTable”)) )
```

Shiny Features

- reactive/observe/isolate functions
- Hover/click buttons
- Debug codes

Summary

- rbokeh and ggvis are easy to learn. To create more complex figures, need to combine with shiny.
- The syntax for shiny is more complex. But, once you master them, it give you a lot of power.

Reference:

- rbokeh:
  - http://hafen.github.io/rbokeh/
- ggvis:
  - https://ggvis.rstudio.com/
  - http://papacochon.com/2015/10/07/Codage-8-ggvis/
- shiny:
  - Tutorial:
    - https://shiny.rstudio.com/tutorial/
  - 2016 Shiny Developer Conference Videos
    - https://www.rstudio.com/resources/webinars/shiny-developer-conference/
  - Shiny Cheat Sheet:
    - https://www.rstudio.com/wp-content/uploads/2016/01/shiny-cheatsheet.pdf