

A Shiny-er future?

*R Shiny as a data exploration
& visualization tool*

Apratim Mitra

Bioinformatics and Scientific Programming Core

NICHD/NIH

NIH BYOB 04-16-2020

Apratim Mitra

- Bioinformatics scientist, NICHD/NIH
- Bioinformatics and Scientific Programming Core
 - <https://bioinformatics.nichd.nih.gov>
- Contact:
 - Email: apratim.mitra@nih.gov
 - Github: <http://github.com/mitraak>

What we do



- Bioinformatic ‘swiss-army knife’
- Work with many different labs at the institute
- Collaborative model - involved with (almost) every step of the project
 - Understand the biology (‘speak the language’)
 - Experimental design
 - Data analysis and management
 - Hypothesis generation and testing
 - Reporting, interpretation, visualization
 - Figure generation, manuscript writing
- Non-standard roles
 - Develop new analysis workflows/tools
 - Learn to use (sometimes undocumented) software
 - <add your request here>

Typical workflow

- Have meeting with scientists to discuss goals and experimental plan
- Get raw data from sequencing core
- Data QC and analysis: *snakemake* or specific tools
 - <https://github.com/lcdb/lcdb-wf>
- Downstream analysis: *R*, *python*
- Reports, follow up analyses, experiments

Reporting tool: Rmarkdown

- Annotate results of analyses
- Add observations or notes that we think might be helpful in interpretation
- Unified way of organizing & navigating complex results
- Version control of document helps organize subsequent updates
- Code is embedded if someone wants to learn the underlying commands
- Example
 - <https://github.com/lcdb/lcdb-wf/blob/master/workflows/rnaseq/downstream/rnaseq.Rmd>

Common questions we get

- How many replicates do we need?
- What kind of controls should we perform?
- Did we get enough reads?
- Is the signal-to-noise ratio good enough?

Experimental design

Data QC

- How many genes are changed due to the treatment?
- What is the p-value threshold?
- Should we use a fold-change cutoff?
- Can you label my favorite genes in this plot?
- What pathways are enriched?
- What genes are coexpressed?

Downstream
Analysis

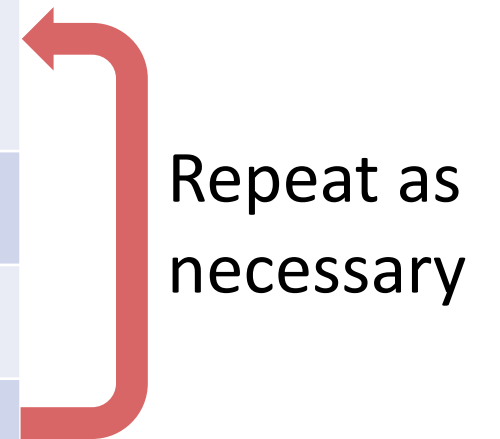
Many back and forth discussions

- Many requests are relatively trivial to perform
 - E.g. subsetting operations (change thresholds), tweak plots (colors, labels)
- Bottleneck in the process
 - Need for quick turnaround
 - Need for precision and fine tuning



Typical timeline

Location	Step	Time
Us	Do the analysis, send results	~ 0.5-1 week
Collaborator	Request to tweak analysis e.g. change p-value threshold	1-2 days
Us	Queue request, rerun, resend	1-2 days
Collaborator	Request to tweak plot	1 day
Us	Queue request, rerun, resend	1-2 days



Time quickly adds up

How to bridge the gap?

- Need an efficient solution
 - Allow biologists to explore data without having to depend on us
 - Allow different options to tweak analysis/results, e.g. visualization
- Rmarkdown is great for many things
 - But it is a static format
- R Shiny can make this possible

What is R Shiny?

- Flexible framework to make interactive web apps from R
- Requires the 'shiny' R package
- Well integrated with RStudio
- Three basic components
 - User interface object
 - Server function
 - Call to shinyApp() function
- Many built in examples, e.g.
 - > runExample('01_hello')
 - > runExample('04_mpg')

```
library(shiny)

# Define UI ----
ui <- fluidPage(

)

# Define server logic ----
server <- function(input, output) {

}

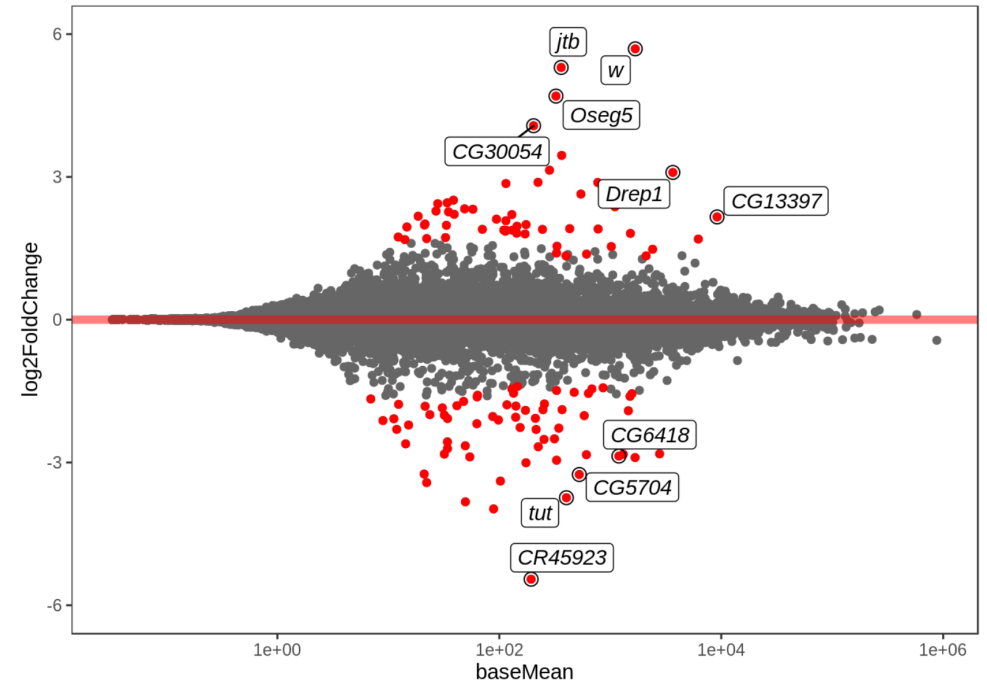
# Run the app ----
shinyApp(ui = ui, server = server)
```

How can it fit our needs?

- Many plots are relatively straightforward to make
 - Exposing function parameters would make it accessible
- Several analyses create unified (large) data structures
 - Native R objects can be saved to file (RDS)
 - These objects can be reused to regenerate plots or rerun analyses

Shiny apps

- Customizable MA plot
 - <http://github.com/mitraak/plotma-shiny>
- Tool for functional enrichment analysis
 - Multiple databases: Gene ontology, KEGG pathways
 - Multiple visualization options: dotplot, emapplot, cnetplot
 - <http://github.com/mitraak/clusterprofiler-ui-shiny>



App deployment

- Share code with collaborator
 - E.g. github, email
 - Needs some familiarity with R, RStudio
 - But well-documented code can help
- External hosting services
 - shinyapps.io, RStudio connect – both integrated with RStudio
- Standalone app builds (?)

External hosting services

- shinyapps.io: <http://shinyapps.io>
 - You can host your apps for free in the basic plan
 - Limited number of active apps (5)
 - 25 free hours per month
 - No authentication
- RStudio Connect: <http://rstudio.com/products/connect>
 - Enterprise solution
 - Base level – 20 users, \$15k/year
 - If hosted on own hardware, no limits on usage
 - Password authentication

Standalone app builds

- Blog post about deploying standalone shiny apps
 - <https://www.travishinkelman.com/posts/deploy-shiny-electron/>
 - <https://github.com/dirkschumacher/r-shiny-electron>
- Template to build shiny app using electron
 - <https://www.electronjs.org/>
 - Architecture-specific build
- But, you're on your own ...



Tips

- Start small
- Code versioning is your friend
- R or package version updates might break things
- Test, test, test
- Resources
 - <https://shiny.rstudio.com/tutorial/>

Conclusions

- R Shiny shows promise
- Easy to use, simple to set up
- Saved native R objects can open up possibilities
 - Explore, rerun, replot
- RStudio integration can facilitate code exchange
- Free public hosting is limited (but free!)
- Large scale public hosting can be expensive
 - But might make sense at an institute level