FMI
Friedrich Miescher Institute
for Biomedical Research

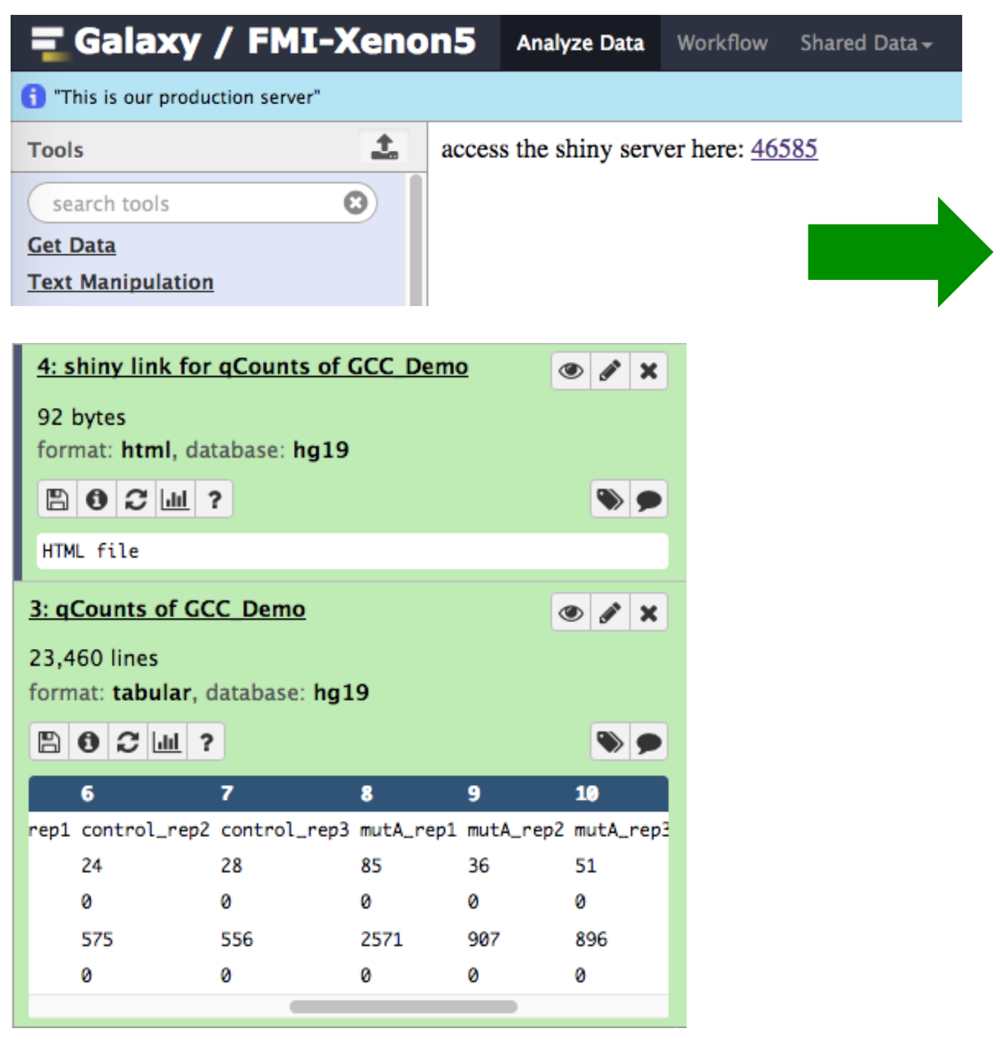
Galaxy2Shiny2Galaxy

Combining Galaxy with Shiny (and doing other bad things)

Hans-Rudolf Hotz, Hélène Royo*, Michael B. Stadler
Friedrich Miescher Institute for Biomedical Research, Basel, Switzerland
* current address: Roche Innovation Center, Basel, Switzerland

We have developed a new way of combining Galaxy with the Shiny framework which does not rely on Docker. Each time a data set (e.g.: a count table from an RNA-Seq experiment) is created in Galaxy, a new app is set up on a local Shiny Server for interactive data exploration. A variety of different plots can be generated and data points of interest can be selected. If a plot or a set of data points is downloaded, the Shiny server stores the visualization (or data table) back into the original Galaxy history with the help of the Galaxy API. At the same time, each step is tracked in a log file on the Shiny Server. The Shiny app is not restricted/protected by the authentication used in Galaxy, and can therefore easily be shared with colleagues not using Galaxy.
We have implemented this combination for two of our R/Bioconductor based NGS tools: 'QuasR::qCount()' and 'edgeR'. Though, the Galaxy tool can be written in any language. A simple bash script is used in the example provided on github (https://github.com/hrhotz/galaxy2shiny2galaxy).

A simple HTML page with the URL to the Shiny server is created as a second data set.



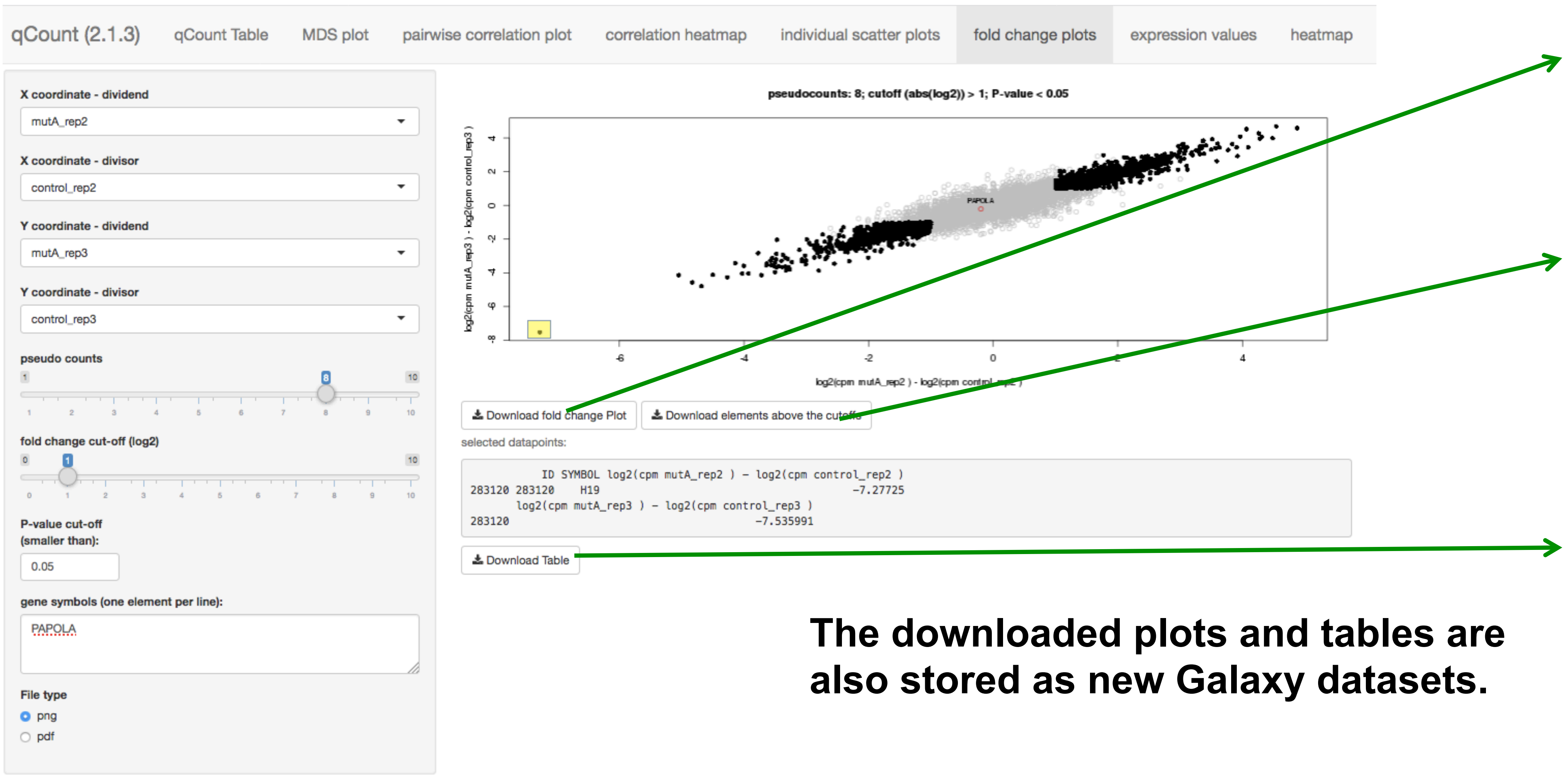
"This is our production server"

access the shiny server here: 46585

4: shiny link for qCounts of GCC_Demo
92 bytes
format: html, database: hg19
HTML file

3: qCounts of GCC_Demo
23,460 lines
format: tabular, database: hg19

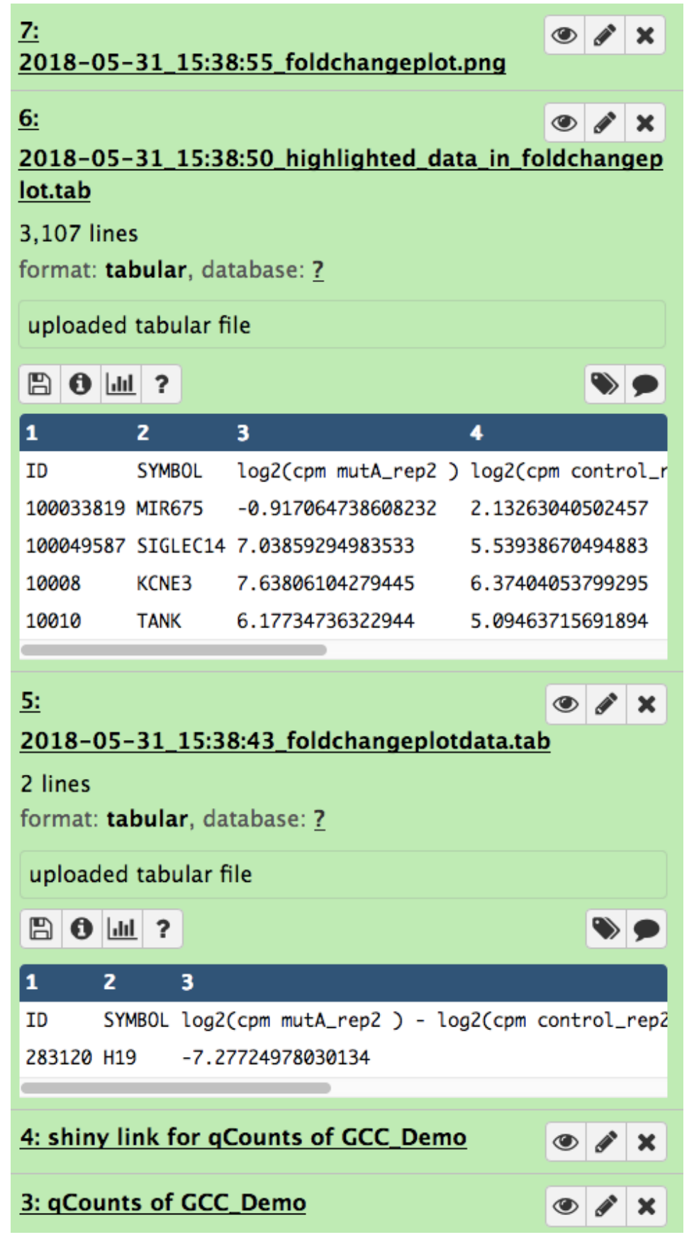
```
6     7            8            9          10
rep1  control_rep2 control_rep3 mutA_rep1  mutA_rep2 mutA_rep3
24    28           85           36         51
0     0            0            0          0
575   556          2571         907        896
0     0            0            0          0
```


qCount (2.1.3)
qCount Table
MDS plot
pairwise correlation plot
correlation heatmap
individual scatter plots
fold change plots
expression values
heatmap
pseudocounts: 8; cutoff (abs(log2)) > 1; P-value < 0.05
log2(cpm mutA_rep3) - log2(cpm control_rep3)
log2(cpm mutA_rep2) - log2(cpm control_rep2)
PAPOLA
Download fold change Plot
Download elements above the cutoffs
Download Table

X coordinate - dividend: mutA_rep2
X coordinate - divisor: control_rep2
Y coordinate - dividend: mutA_rep3
Y coordinate - divisor: control_rep3
pseudo counts: 8
fold change cut-off (log2): 1
P-value cut-off (smaller than): 0.05
gene symbols (one element per line): PAPOLA
File type
● png
○ pdf

selected datapoints:

```
          ID SYMBOL log2(cpm mutA_rep2 ) - log2(cpm control_rep2 )
283120 283120    H19                                      -7.27725
       log2(cpm mutA_rep3 ) - log2(cpm control_rep3 )
283120                                      -7.535991
```



7: 2018-05-31_15:38:55_foldchangeplot.png

6: 2018-05-31_15:38:50_highlighted_data_in_foldchangeplot.tab
3,107 lines
format: tabular, database: ?
uploaded tabular file

```
1         2        3                    4
ID        SYMBOL   log2(cpm mutA_rep2 ) log2(cpm control_r
100033819 MIR675   -0.917064738608232   2.13263040502457
100049587 SIGLEC14 7.03859294983533     5.53938670494883
10008     KCNE3    7.63806104279445     6.37404053799295
10010     TANK     6.17734736322944     5.09463715691894
```

5: 2018-05-31_15:38:43_foldchangeplotdata.tab
2 lines
format: tabular, database: ?
uploaded tabular file

```
1      2      3
ID     SYMBOL log2(cpm mutA_rep2 ) - log2(cpm control_rep2
283120 H19    -7.27724978030134
```

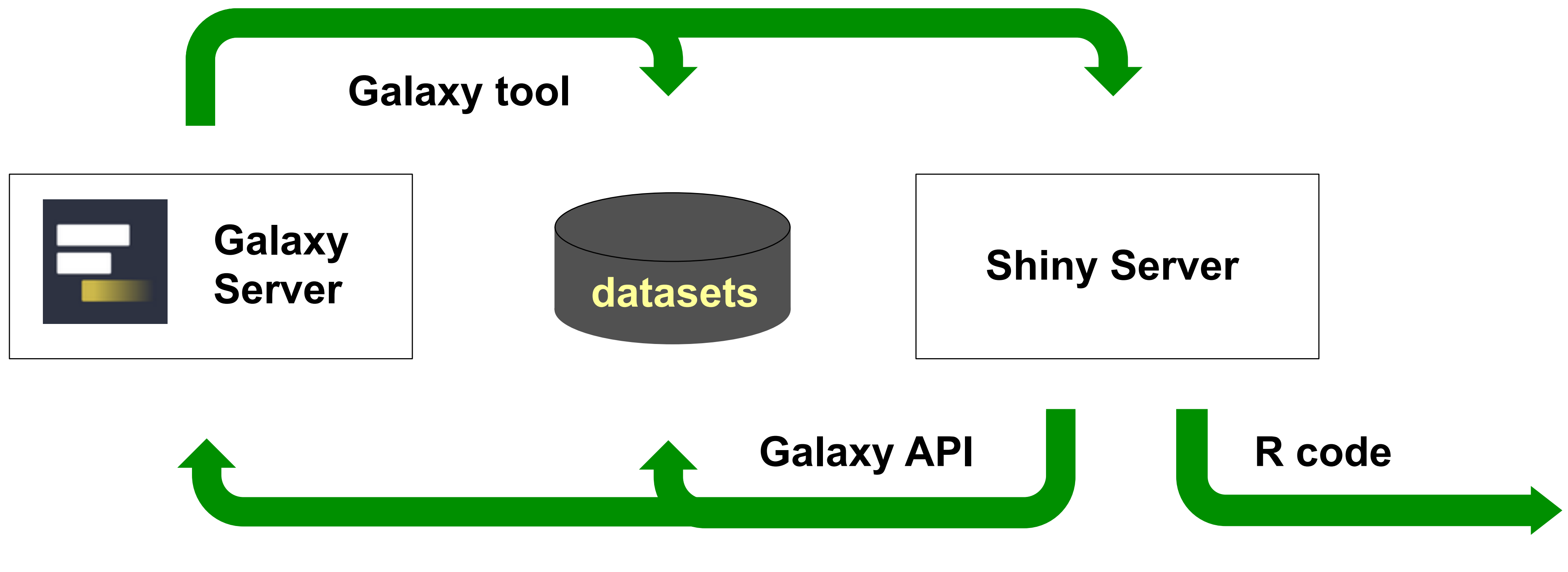
4: shiny link for qCounts of GCC_Demo

3: qCounts of GCC_Demo

The downloaded plots and tables are also stored as new Galaxy datasets.

How does it work

The existing R script generating the count table (based on the QuasR Bioconductor package) has been modified to create a new Shiny app. The galaxy job id is used to generate a unique directory name (and URL) on the Shiny Server.



Clicking on any 'download' buttons in the Shiny app triggers a call to the Galaxy API to store the dataset using the encoded history id and the API key from an admin account.

directory structure of the Shiny app

`server.R`
`ui.R` } copies of generic scripts to handle data generated by edgeR or QuasR

`qCountTable -> /path/to/dataset_1234.dat`

`log` each interaction in the Shiny app is stored by logging all input variables

`encoded.history.id` required by the Galaxy API and generated from the path of $outputfile

requirements

- The user, the Galaxy Server runs as, needs to have write access to the apps directory of the Shiny Server
- The apps directory of the Shiny Server needs to be mounted on the server(s) Galaxy runs on
- The dataset directory of the Galaxy Server needs to be mounted on the server Shiny runs on
- You need to be an admin for your Galaxy installation, and need to have write access to the Galaxy code
- You need write access to the Galaxy PostgreSQL database

resources

https://github.com/hrhotz/galaxy2shiny2galaxy

https://cran.r-project.org/web/packages/shiny/
https://www.rstudio.com/products/shiny/shiny-server/
https://bioconductor.org/packages/release/bioc/html/QuasR.html
https://bioconductor.org/packages/release/bioc/html/edgeR.html

Special Thanks to Helena Rasche for the utility script to encode and decode IDs from the Galaxy database: https://github.com/galaxyproject/galaxy/pull/3622