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. 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.

requirement

- 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

resource

https://github.com/hrhotz/galaxy2shiny2galaxy
https://cran.r-project.org/web/packages/shiny/
https://www.rstudio.com/products/shiny/shiny-server/

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