Jupyter Notebooks for Performing & Sharing Bioinformatics Analyses

Jonathan Dursi, CCM
jonathan@dursi.ca
https://github.com/ljdursi/glbio-jupyter-workshop

Healthier Children. A Better World.

SickKids RESEARCH INSTITUTE

The plan for this morning

Run Jupyter (née iPython) notebooks on your laptop, in R

How to perform interactive analyses in a web browser using Jupiter

How does it all work?

Using markdown and latex to format notebooks nicely

"Port" an R bioinformatics workflow from scripts into a Jupyter notebook

Sharing a Jupyter notebook online:

- SageMathCloud
- GitHub and git; nbviewer
- mybinder.org

The plan for this morning

Perfectly reasonable and interesting things to discuss that we ***won't*** be talking about today:

- Combining R and Python: cool and possible but requires a little setup
- Jupyter + Spark, Julia, or any of the other languages supported
- IPython magics
- The zillion R bioinformatics packages that are available
- How to use R, Python

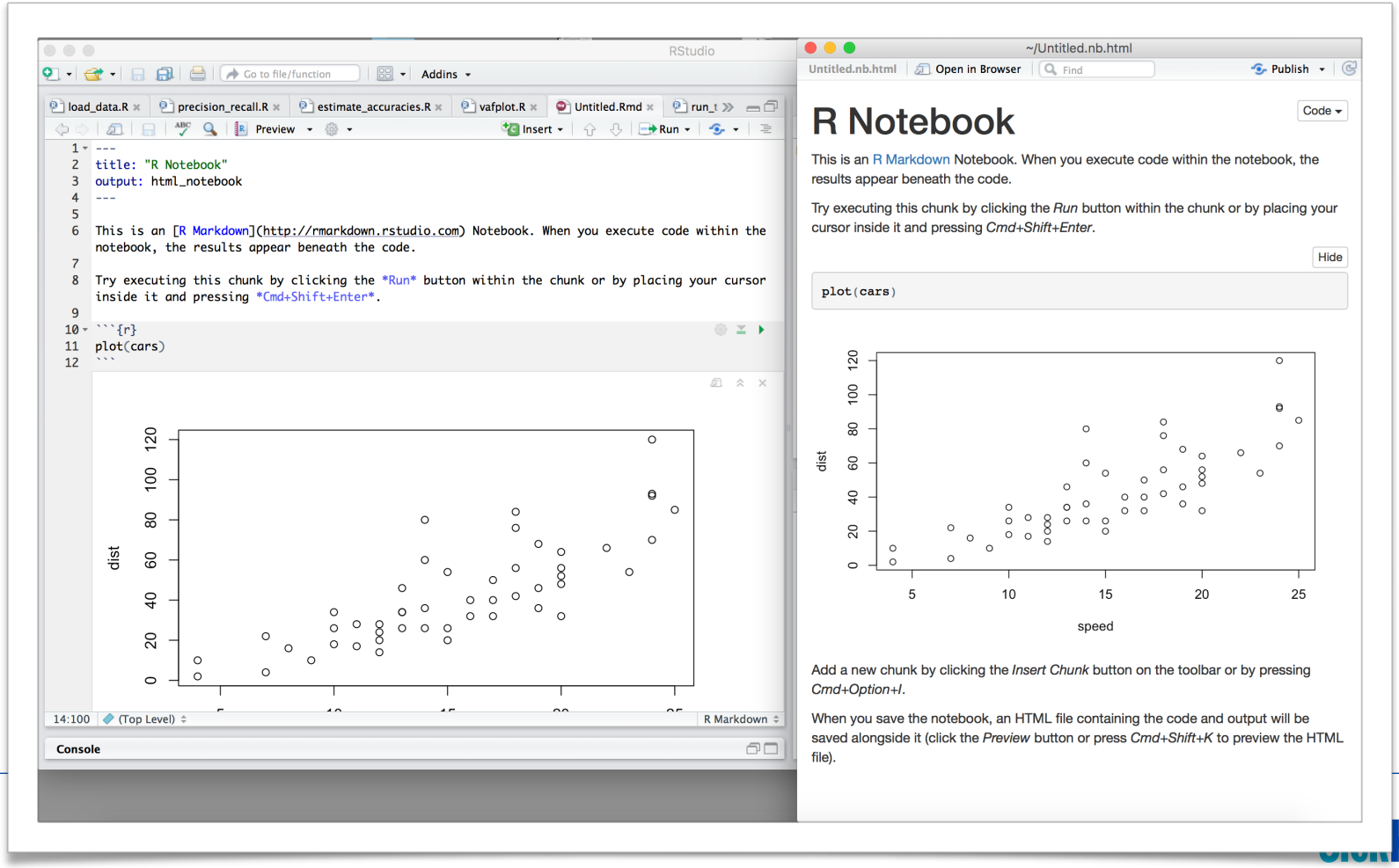
Where Does Jupyter Fit In?

Various tools, particularly for the R environment, exist for sharing computations / analyses:

- Shiny: interactive web applications
- R Markdown:
    - R presentations
    - Replicatable documents
    - Both sharable and runnable but not interactive

Where Does Jupyter Fit In?

Current preview version of RStudio (https://www.rstudio.com/products/rstudio/download/preview/) even has notebooks:



Where Does Jupyter Fit In?

But Jupyter has the following advantages

- Multi-language/framework support
    - Python, Scala, Julia, Haskel, Go, Bash…
    - Spark
- Interactive computation + final presentation
- Several ways to share, either statically or runnable.

Let's start up a notebook

Make a Notebooks directory and start from there

Command Line: Linux, Mac

6. python3.5

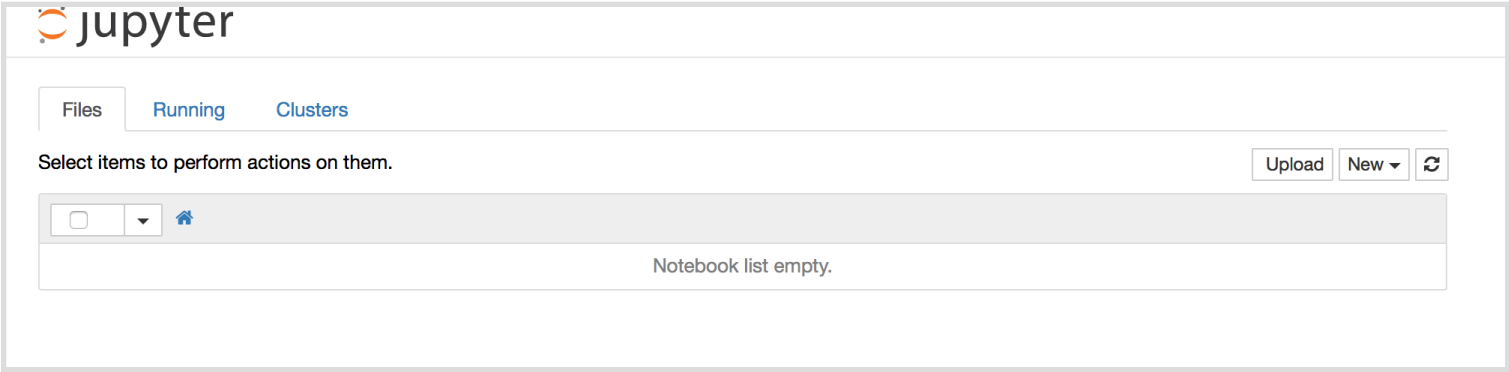
```
Last login: Fri Sep  2 09:22:47 on ttys005
coredump:~ ljdursi$ mkdir ~/Desktop/Notebooks
coredump:~ ljdursi$ cd ~/Desktop/Notebooks
coredump:Notebooks ljdursi$ jupyter notebook
[I 09:23:16.310 NotebookApp] Serving notebooks from local directory: /Users/ljdursi/Desktop/Notebooks
[I 09:23:16.310 NotebookApp] 0 active kernels
[I 09:23:16.310 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 09:23:16.310 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 09:23:17.203 NotebookApp] 404 GET /apple-touch-icon-precomposed.png (::1) 8.45ms referer=None
[W 09:23:17.214 NotebookApp] 404 GET /apple-touch-icon.png (::1) 1.64ms referer=None
```

Let's start up a notebook

Make a Notebooks directory and start from there

Command Line: Linux, Mac

Browser should open (or go to localhost:8888)

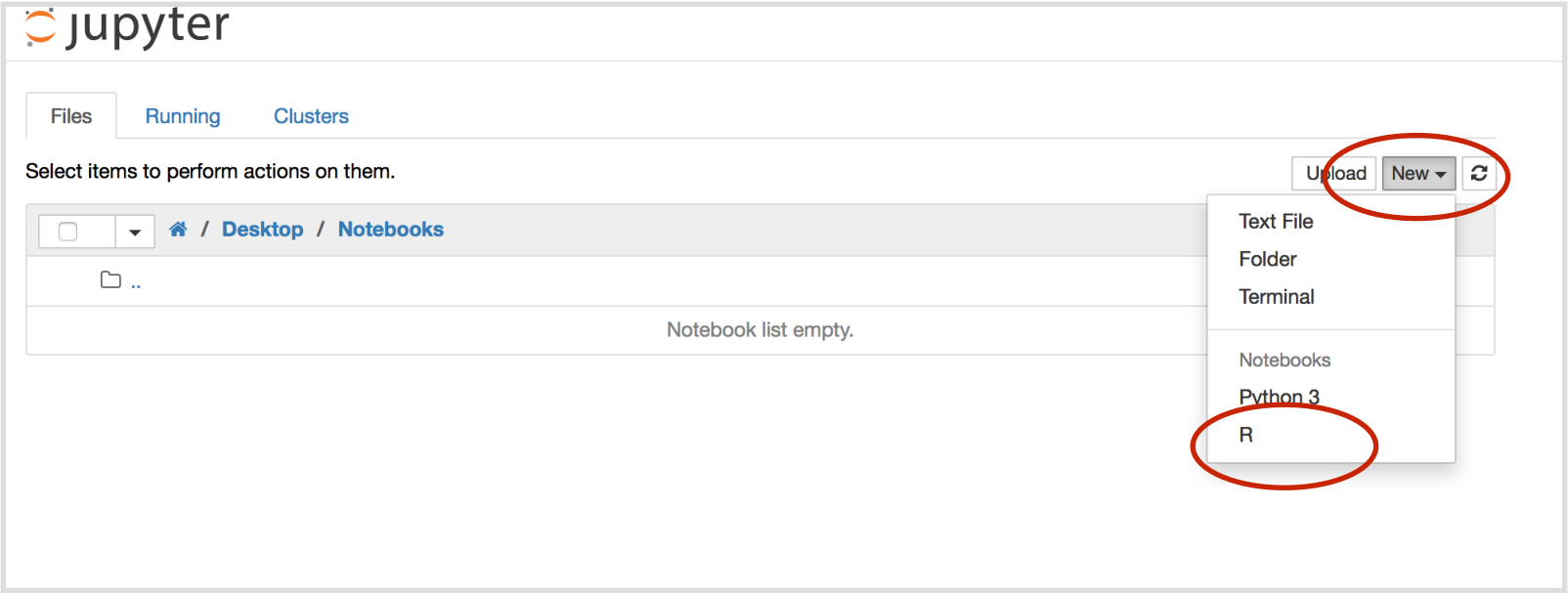


Let's start up a notebook

Or you can start from home (eg, launch from start menu: Win) and navigate folders, and create new folders:

Let's start up a notebook

Let's start up an R notebook



Demo - starting notebook

1-Introduction

How does this all work?

Let's start up an R notebook

notebook server

```
coredump:~ ljdursi$ mkdir ~/Desktop/Notebooks
coredump:~ ljdursi$ cd ~/Desktop/Notebooks/
coredump:Notebooks ljdursi$ jupyter notebook
[I 12:56:20.381 NotebookApp] The port 8888 is already in use, trying another random port.
[I 12:56:20.390 NotebookApp] Serving notebooks from local directory: /Users/ljdursi/Desktop/N
otebooks
[I 12:56:20.391 NotebookApp] 0 active kernels
[I 12:56:20.391 NotebookApp] The Jupyter Notebook is running at: http://localhost:8889/
[I 12:56:20.391 NotebookApp] Use Control-C to stop this server and shut down all kernels (twi
ce to skip confirmation).
```

nb#1

R kernel 1

```
a <- "Hello World!"
x <- 10
y <- 5
```

nb#2

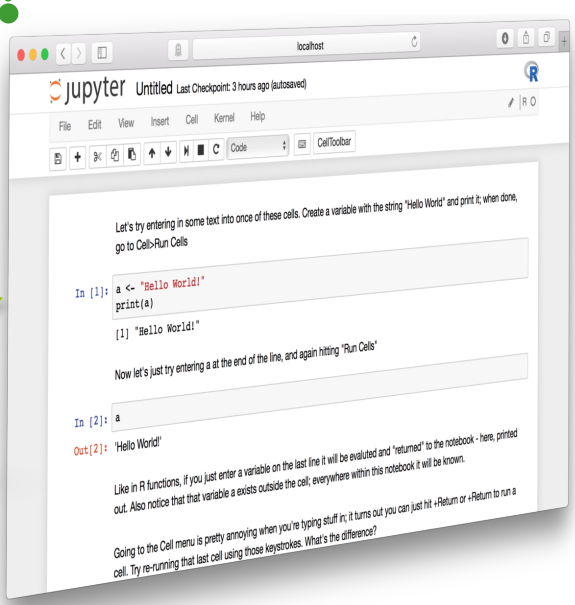
R kernel 2

```
a <- "Goodbye!"
y <- -1
```

nb#3

Py kernel1

```
import numpy as np
x = np.arange(0,10)
y = np.array([1,2])
```

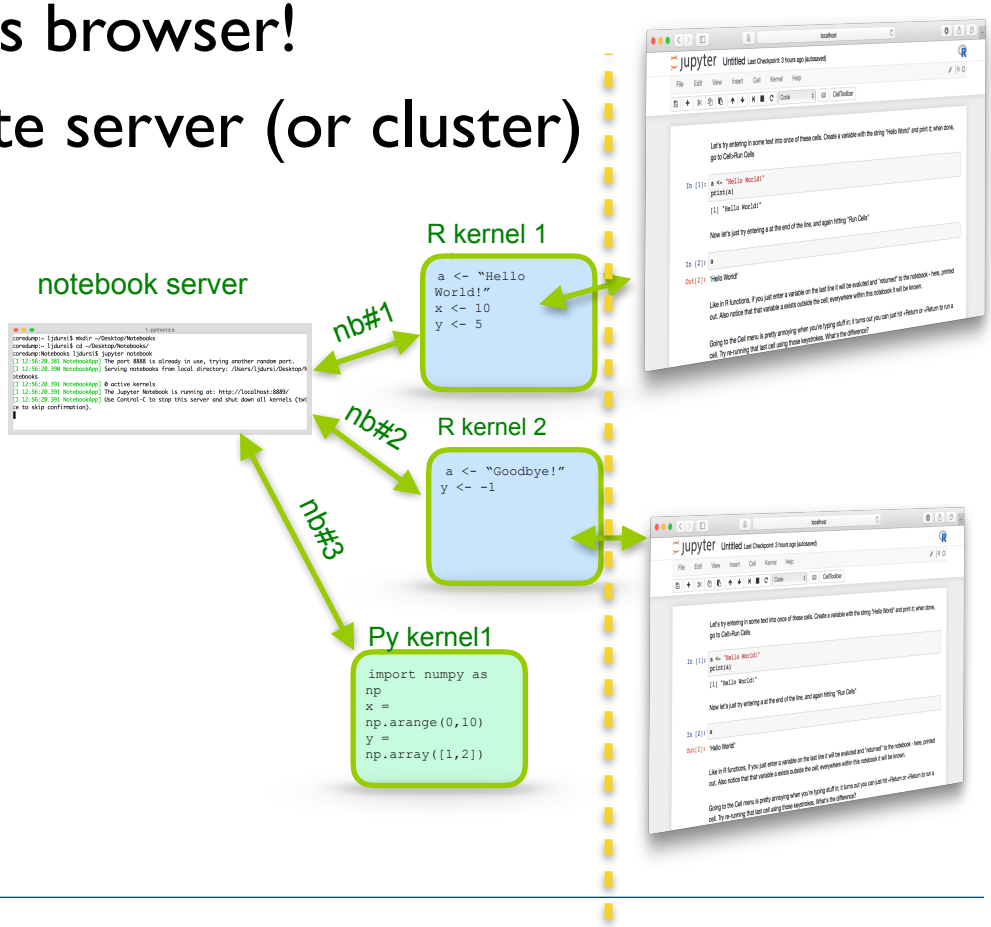


How does this all work?

There's no rule that says the notebook server/kernels has to be on same machine as browser!

Can be running on remote server (or cluster)

AWS demo

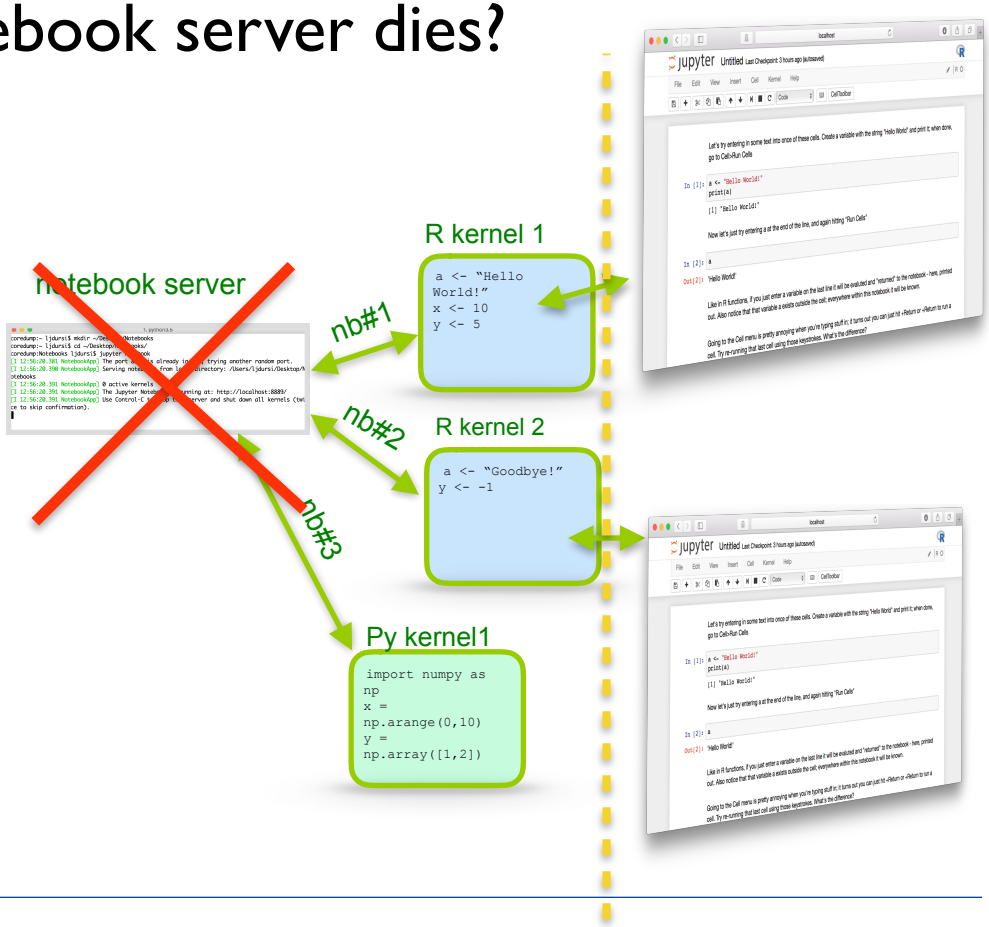


How does this all work?

AWS demo…

What happens if the notebook server dies?

Demo



All input+output in ipynb file

All the input+output is located in the .ipynb file

jupyter Untitled1 Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help

Markdown CellToolbar

```
                                        as.is=TRUE)
```

In [5]:
```
pDataFile = file.path(dataDirectory, "pData.txt")
pData = read.table(pDataFile,row.names=1, header=TRUE, sep="\t")
```

In [6]:
```
head(pData)
```

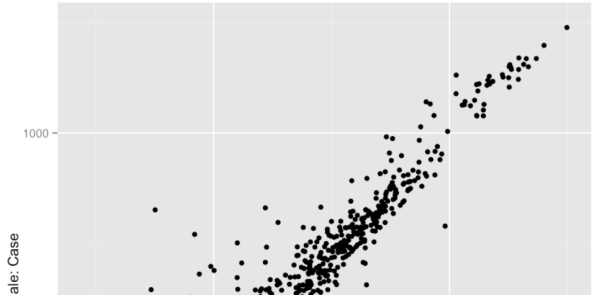
Out[6]:

	gender	type	score
A	Female	Control	0.75
B	Male	Case	0.4
C	Male	Control	0.73
D	Male	Case	0.42
E	Female	Case	0.93
F	Male	Control	0.22

Correlations

What does the correlation between case A and E: a female control and case: look like?

In [7]:
```
library(ggplot2)
ggplot(exprs) + geom_point(aes(x=abs(A), y=abs(E))) + xlab('Female: Control') + ylab('Female: Case') +
    scale_x_log10() + scale_y_log10()
```




```
{
 "cells": [
...
   "outputs": [],
   "source": [
    "data <- read.csv('http://dursi.ca/content/images/GSE37704_featurecounts.csv')\n",
    "metadata <- read.csv('http://dursi.ca/content/images/GSE37704_metadata.csv')"
   ]
  },
...
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "'data.frame':\t19808 obs. of  8 variables:\n",
      " $ ensgene  : Factor w/ 19808 levels \"ENSG00000000003\",..: 15488 19676 19628 19554 19334 15792 16078 15
867 15778 15793 ...\n",
      " $ length   : int  918 718 1982 939 939 3214 5539 3395 2833 3424 ...\n",
      " $ SRR493366: int  0 0 23 0 0 124 1637 120 24 4 ...\n",
      " $ SRR493367: int  0 0 28 0 0 123 1831 153 48 9 ...\n",
      " $ SRR493368: int  0 0 29 0 0 205 2383 180 65 16 ...\n",
      " $ SRR493369: int  0 0 29 0 0 207 1226 236 44 14 ...\n",
      " $ SRR493370: int  0 0 28 0 0 212 1326 255 48 16 ...\n",
      " $ SRR493371: int  0 0 46 0 0 258 1504 357 64 16 ...\n"
     ]
    }

...
   "outputs": [
    {
     "data": {
      "image/png":
      "iVBORw0KGgoAAAANSUhEUgAAA0gAAANICAYAAAD958/bAAAEDWlDQ1BJQ0MgUHJvZmlsZQAAO
      I2NVV1oHFUUPrtzZyMkzlNsNIV0qD8NJQ2TVjShtLp/3d02bpZJNtoi6GT27s6Yyc44M7v9oU9
      FUHwx6psUxL+3gCAo9Q/bPrQvlQol2tQgKD60+INQ6Ium65k7M5lpurHeZe58853vnnvuuWfvB
```

Demo - nb formatting

2-Making Nice Notebooks

Healthier Children. A Better World.

SickKids RESEARCH INSTITUTE

“Hands on”: make a nice notebook

3 - Pretty Notebook

Healthier Children. A Better World.

SickKids RESEARCH INSTITUTE

nbconvert

Once you have a nice notebook, there's a few things you can immediately do with jupyter nbconvert:

--to html

--to slides

demo

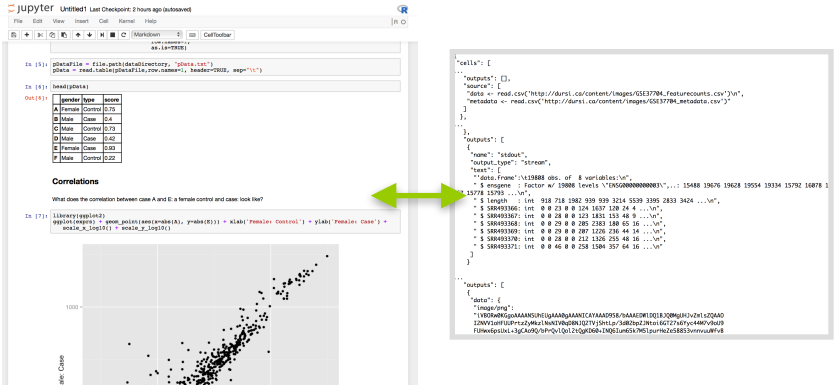
Demo: Moving scripts to notebook

4 - Moving scripts to notebook.ipynb

Sharing

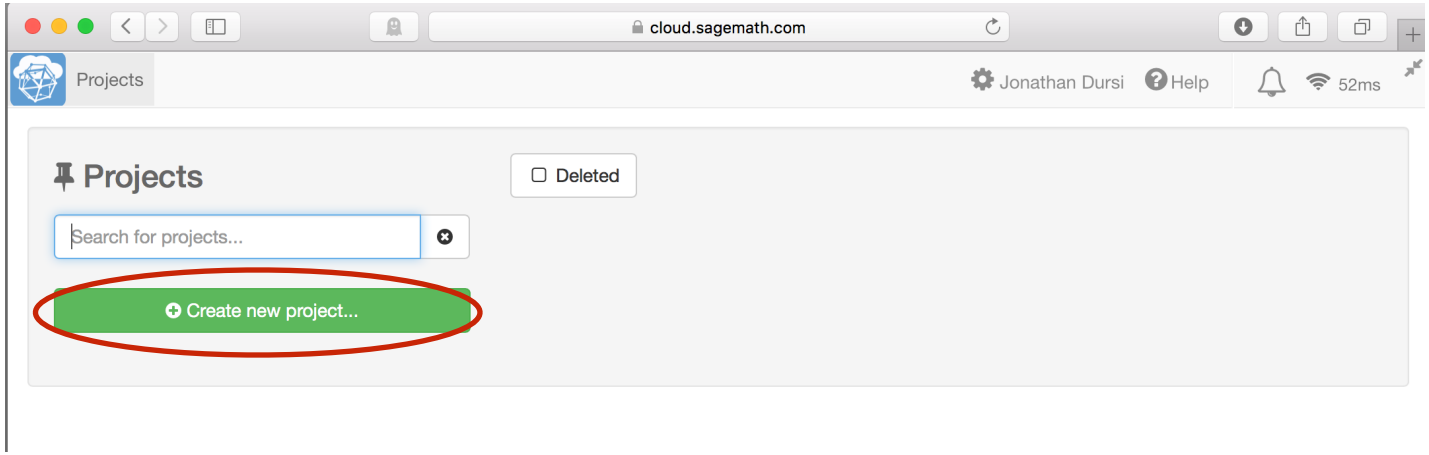
All the input+output is located in the .ipynb file

- As we've seen, Jupiter notebooks can be pretty self-contained
- (May refer to local or remote data files, etc)
- For others to view or interact with, need to find a place willing to host+render the files (to view), or provide the compute to re-run the analyses (to reproduce)
- We'll look at:
    - SageMathCloud
    - GitHub, nbviewer
    - MyBinder



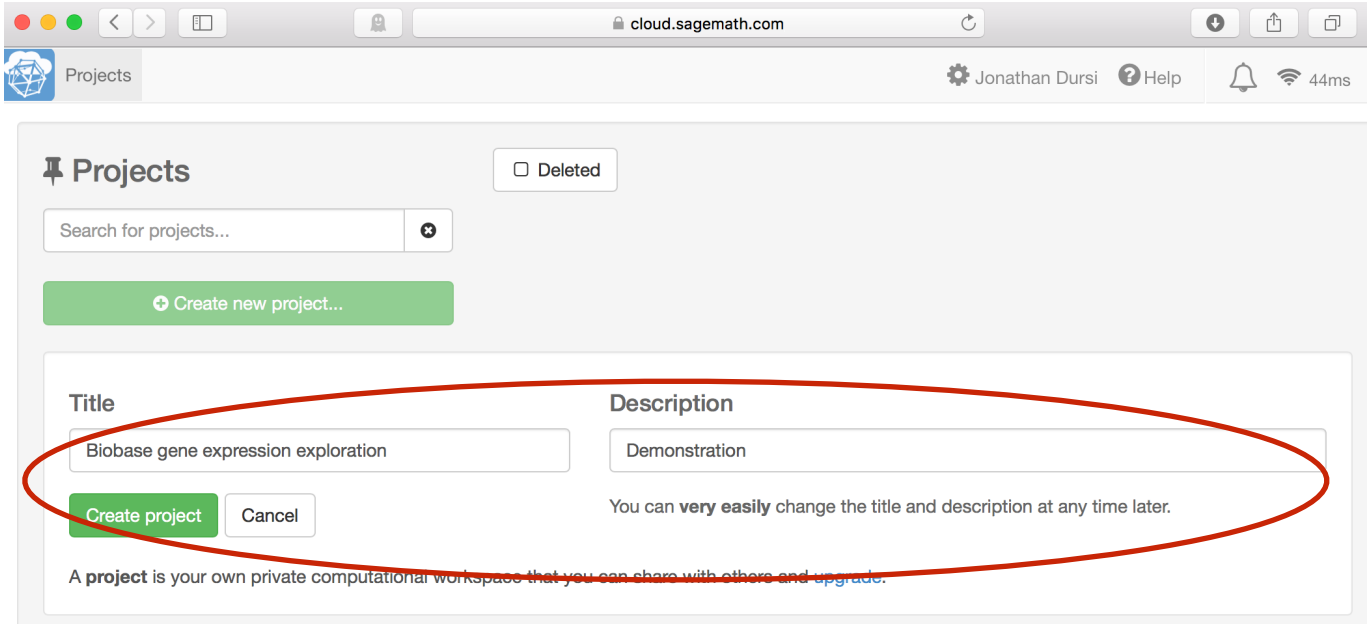
Sharing: SageMathCloud

Log into SageMathCloud (cloud.sagemath.com); create new project



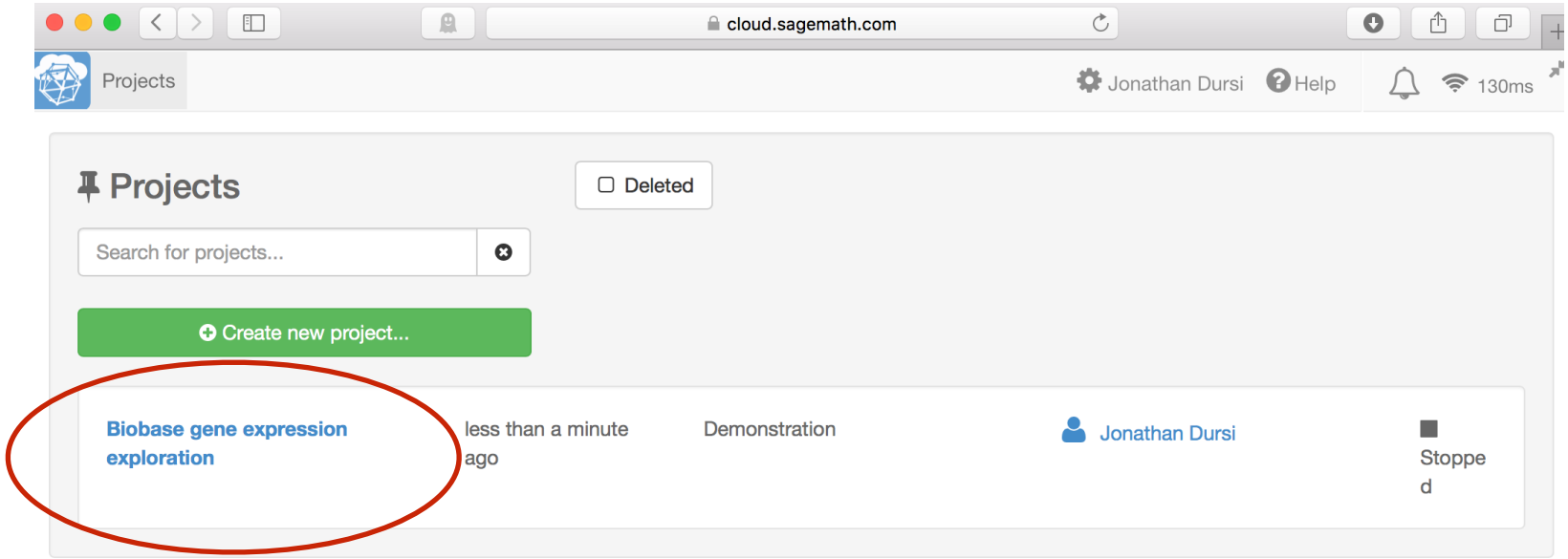
Sharing: SageMathCloud

Log into SageMathCloud (cloud.sagemath.com); create new project



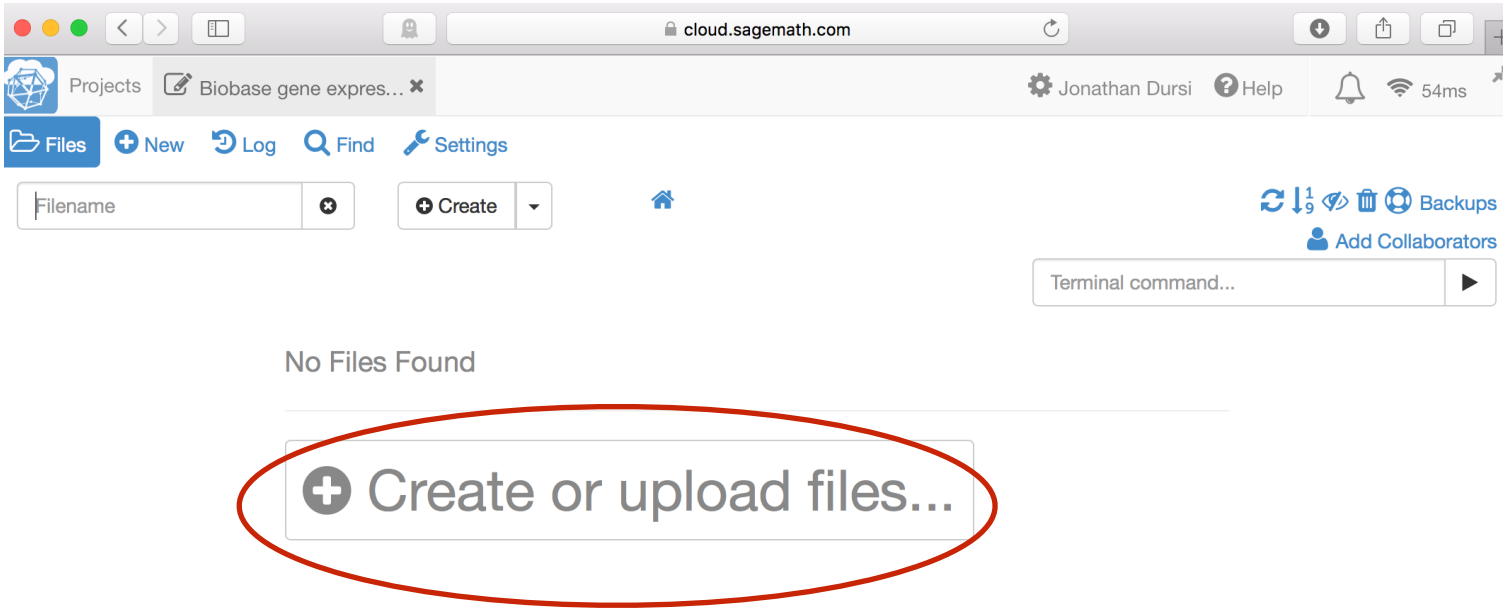
Sharing: SageMathCloud

Then Enter the project…



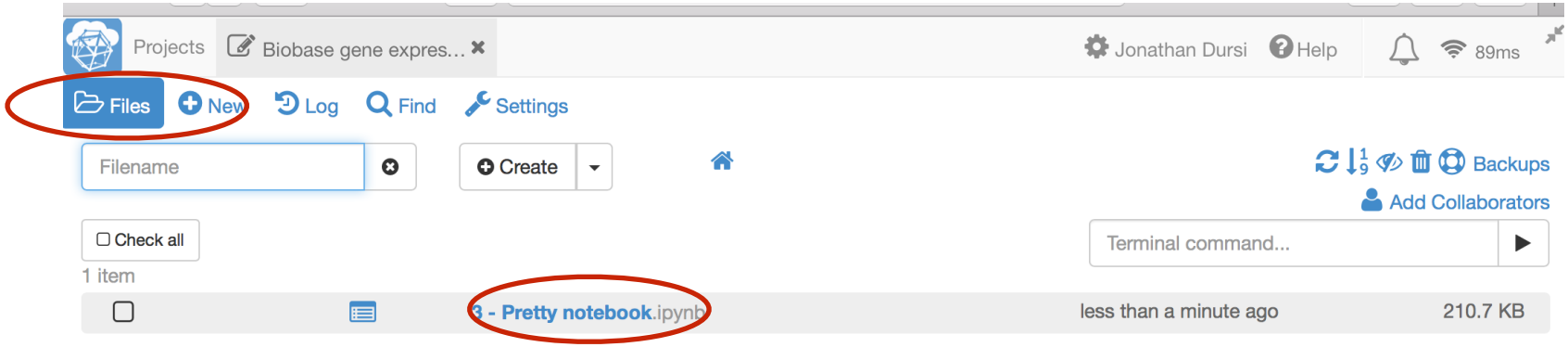
Sharing: SageMathCloud

And upload your notebook



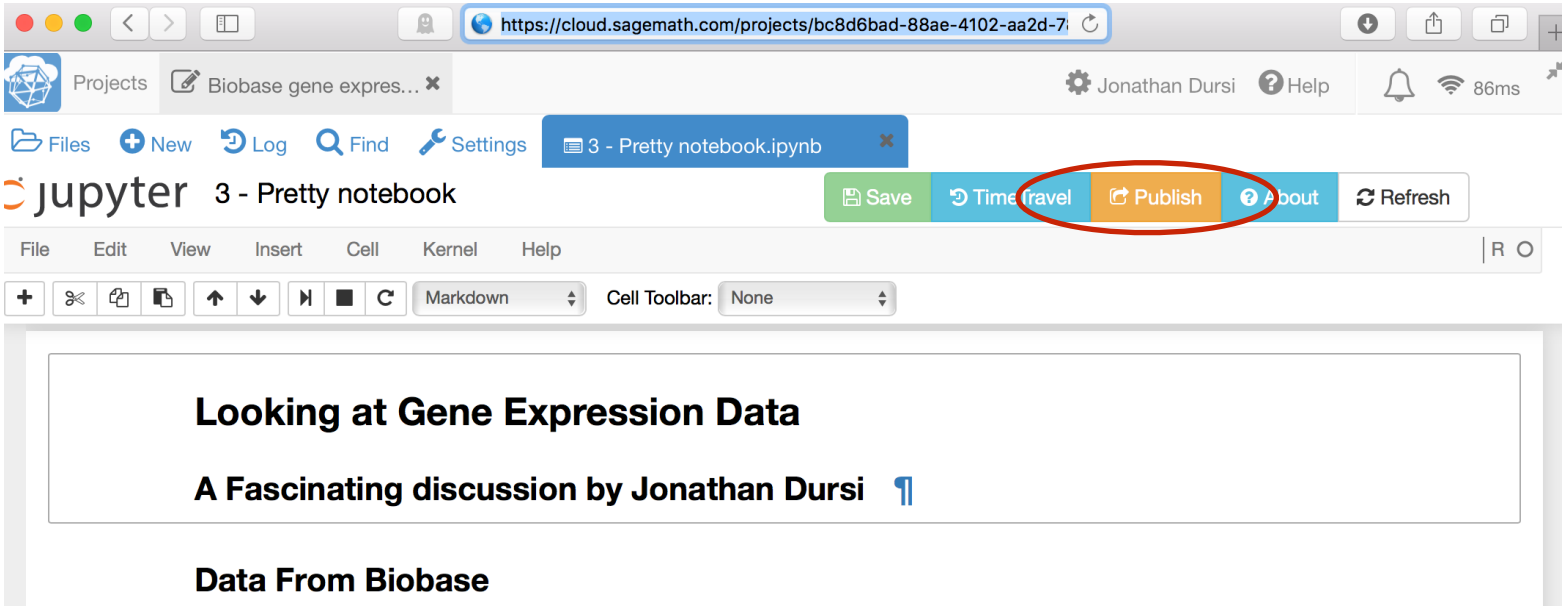
Sharing: SageMathCloud

Then go back to files and launch your notebook



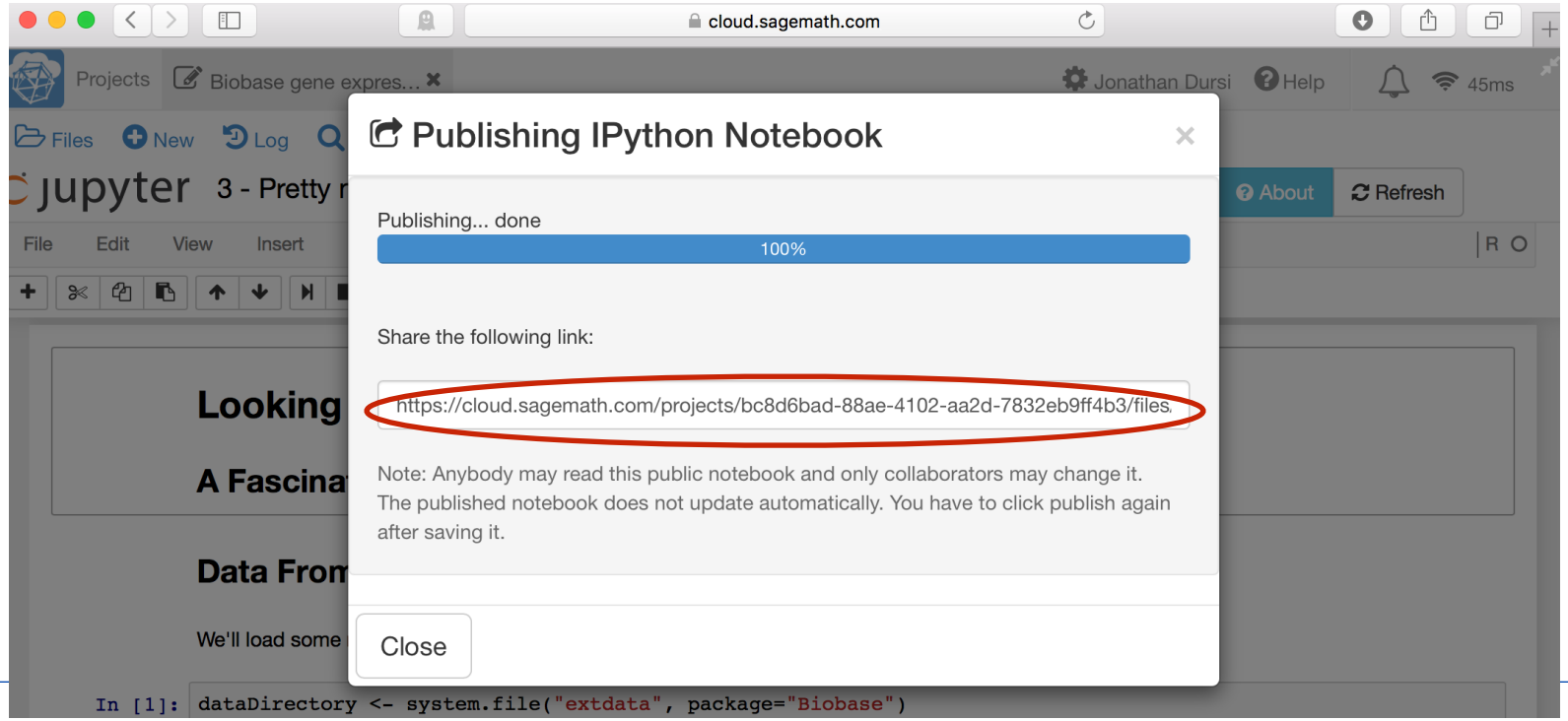
Sharing: SageMathCloud

…and click publish



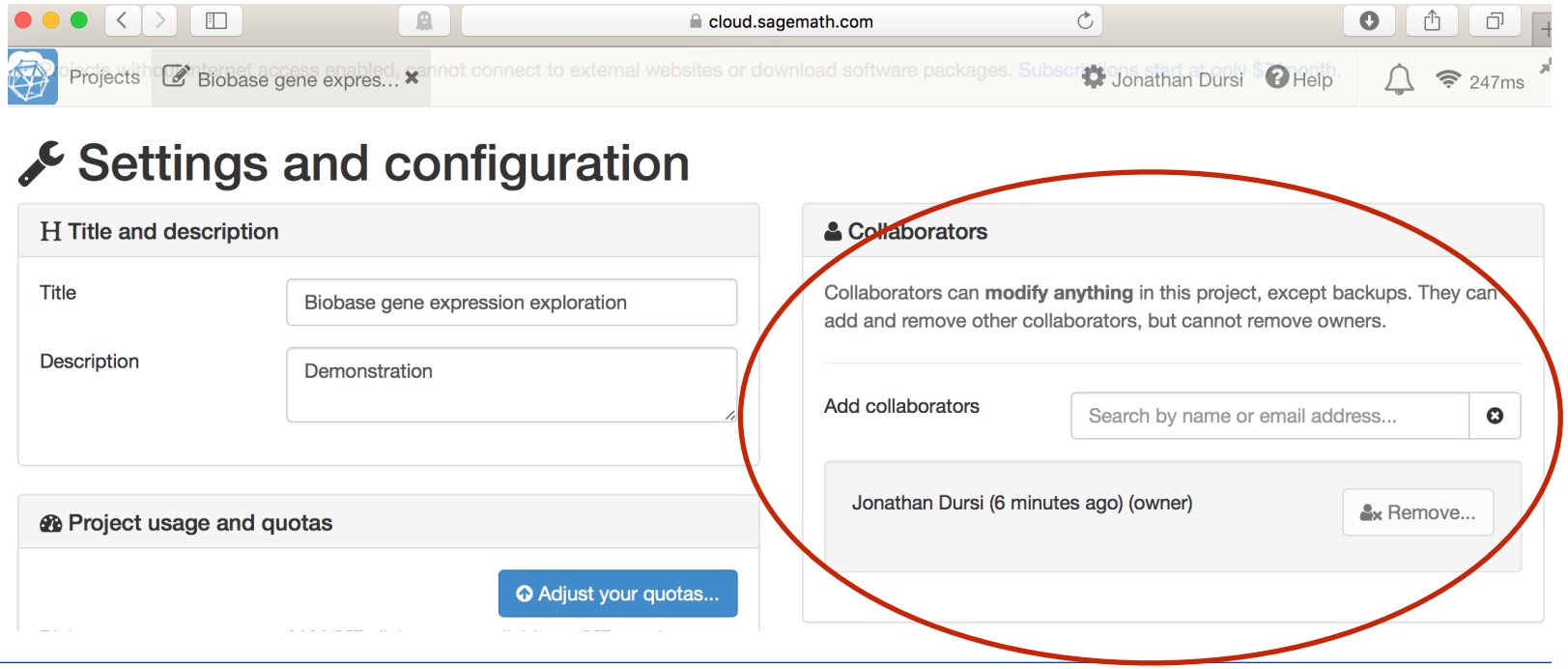
Sharing: SageMathCloud

You can now share it with anyone to *view*…



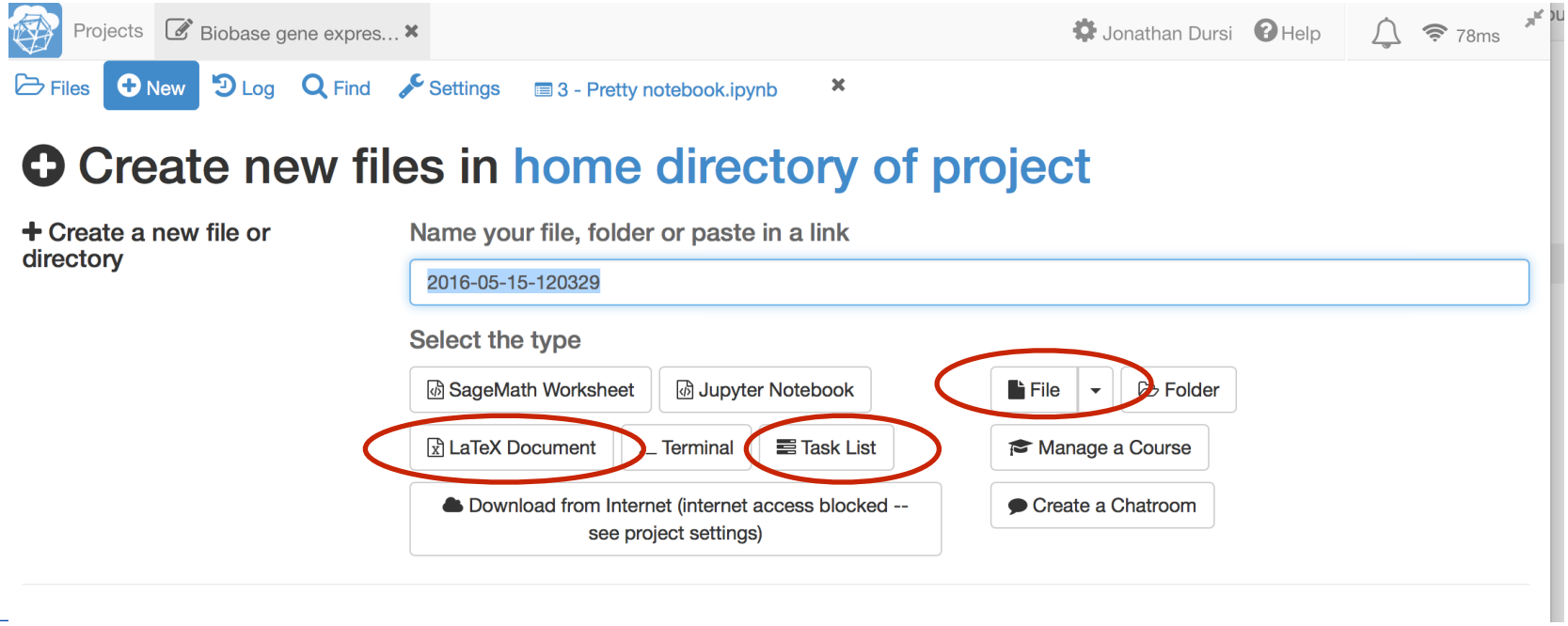
Sharing: SageMathCloud

…Or add collaborators to edit.



Sharing: SageMathCloud

Can share to-do lists, data files, writeups: but many features require $7/mo subscription



GitHub

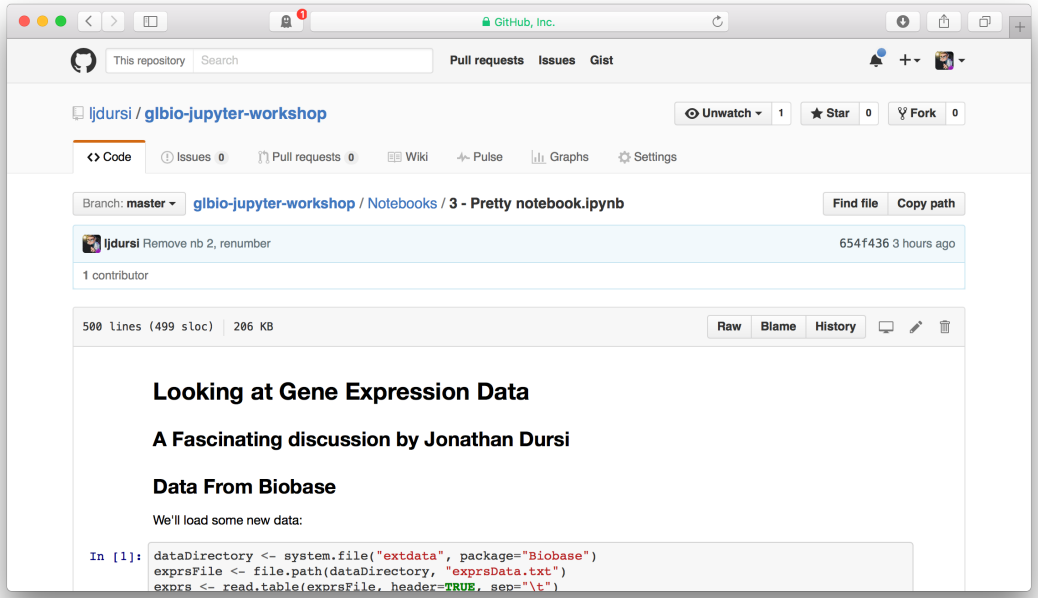
Widely used for collaborating on scientific software

Will render uploaded .ipynb files automatically

Can get unlimited private repositories for researchers:

https://education.github.com

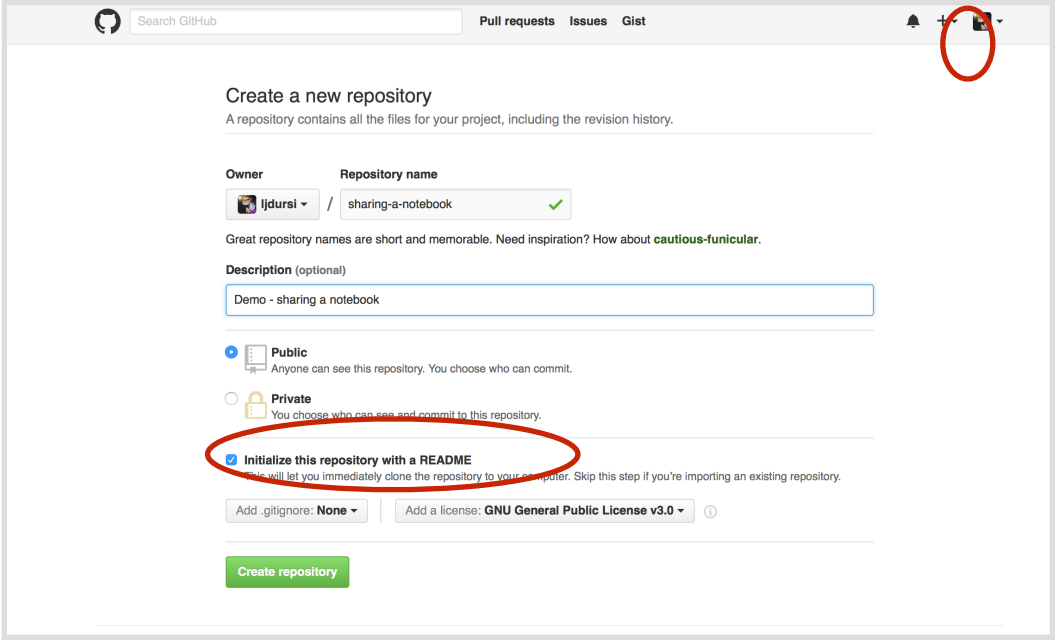
plus other stuff for students



Sharing in github

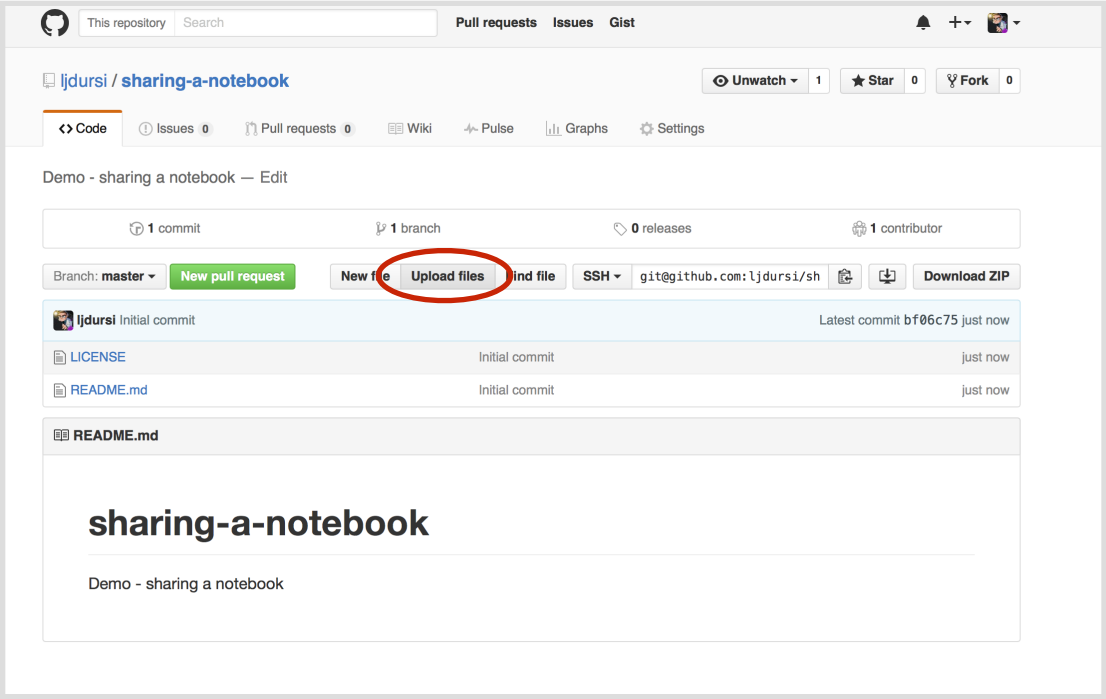
Easy way: go to github.com and create a repo:

Fill in the details and initialize a README



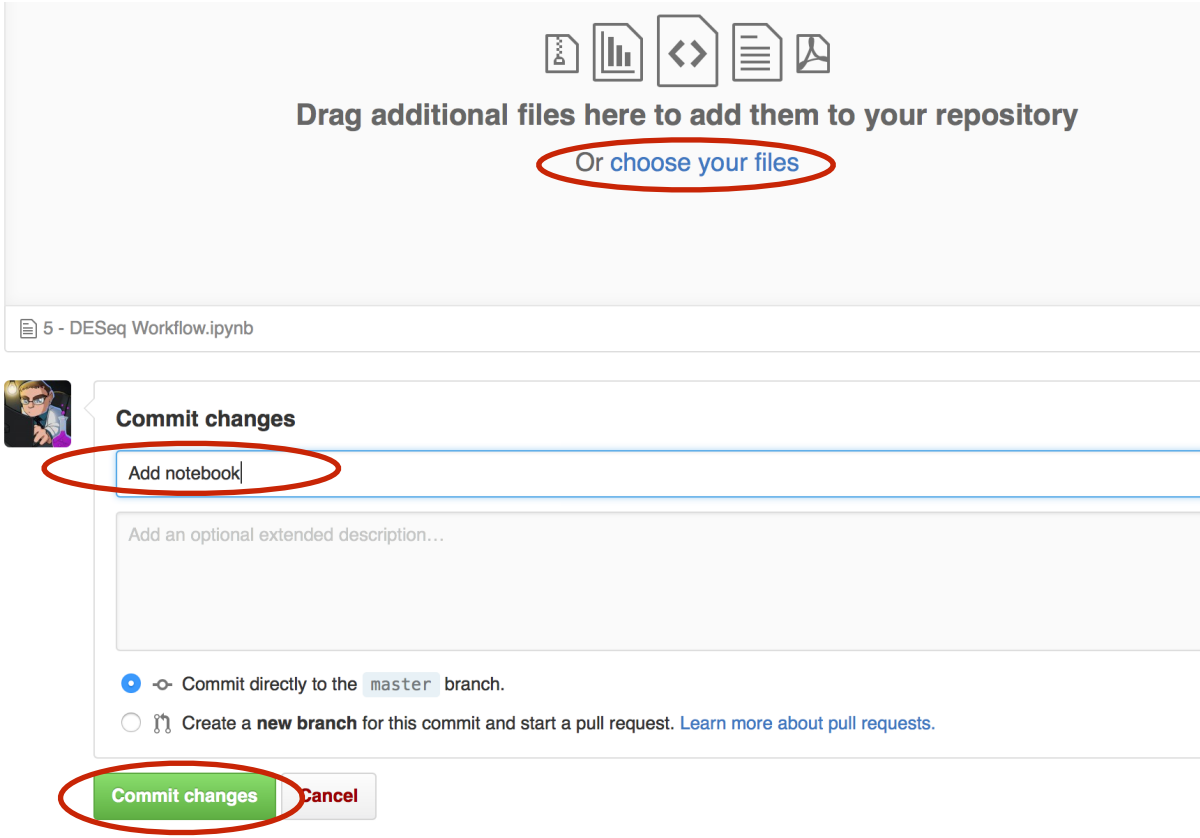
Sharing in github

Now upload your notebook:



Sharing in github

Now upload your notebook:



Sharing in github

And just view the file in the repo

Branch: master ▾ sharing-a-notebook / 5 - DESeq Workflow.ipynb Find file Copy path

ljdursi Add notebook 7e2fb1f 20 seconds ago

1 contributor

803 lines (802 sloc) | 733 KB Raw Blame History

RNA-seq analysis with DESeq2

Largely from Stephen Turner, @genetics_blog

Using data from GSE37704, with processed data available on Figshare DOI: 10.6084/m9.figshare.1601975. This dataset has six samples from GSE37704, where expression was quantified by either:

- Mapping to to GRCh38 using STAR then counting reads mapped to genes with featureCounts under the union-intersection model, or
- Alignment-free quantification using Sailfish, summarized at the gene level using the GRCh38 GTFile. Both datasets are restricted to protein-coding genes only.

We're using the Sailfish data.

This example comes from a gist template for DESeq analyis and a blog post, both by Stephen Turner.

First we pull in the data from a remote copy:

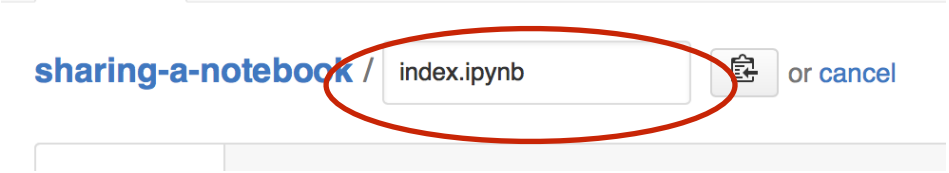
In [1]:
```
countdata <- read.csv("http://www.dursi.ca/content/images/GSE37704_featurecounts.csv", header=TRUE
, row.names=1)
coldata <- read.csv("http://www.dursi.ca/content/images/GSE37704_metadata.csv", header=TRUE,
row.names=1)
```

Sharing in github

Let's give it a better name; click on the pencil to edit the file:

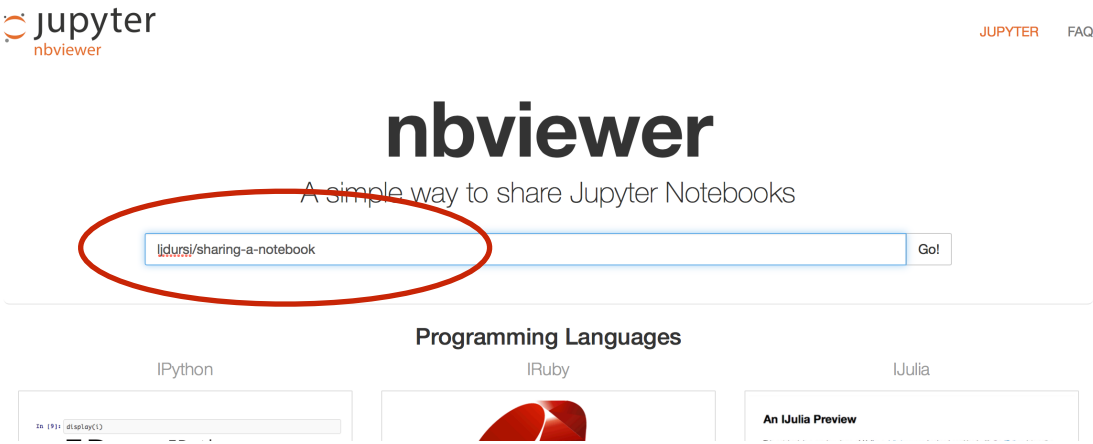
Sharing in github

and then rename it to index.ipynb



nbviewer.jupyter.org

- Once the file is hosted somewhere (github, or anywhere) nbviewer can view it
- Enter name of repo:
  ljdursi/sharing-a-notebook



nbviewer.jupyter.org

- Produces slightly nicer results
- You can share the link it goes to directly, or download the notebook

jupyter nbviewer

JUPYTER FAQ

sharing-a-notebook / index.ipynb

RNA-seq analysis with DESeq2

Largely from Stephen Turner, @genetics_blog

Using data from GSE37704, with processed data available on Figshare DOI: 10.6084/m9.figshare.1601975. This dataset has six samples from GSE37704, where expression was quantified by either:

- Mapping to to GRCh38 using STAR then counting reads mapped to genes with featureCounts under the union-intersection model, or
- Alignment-free quantification using Sailfish, summarized at the gene level using the GRCh38 GTFile. Both datasets are restricted to protein-coding genes only.

We're using the Sailfish data.

This example comes from a gist template for DESeq analyis and a blog post, both by Stephen Turner.

First we pull in the data from a remote copy:

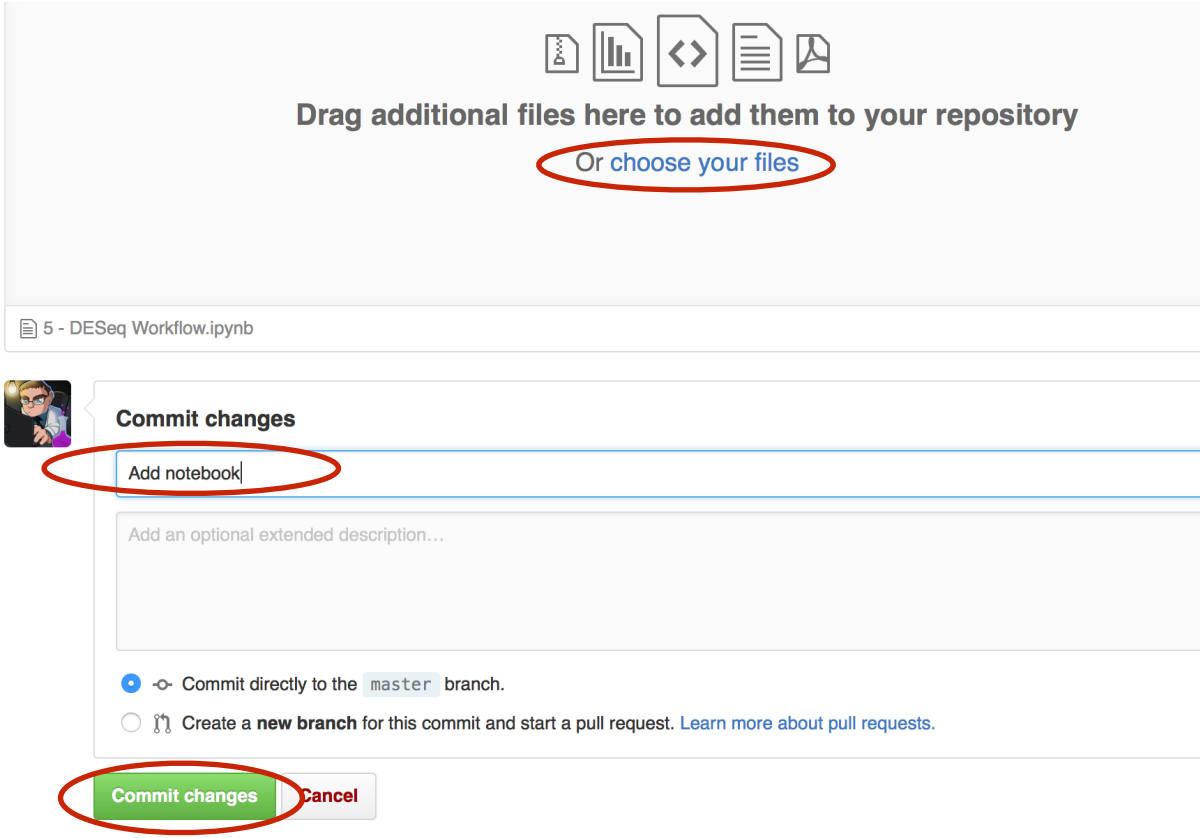
In [1]:
```
countdata <- read.csv("http://www.dursi.ca/content/images/GSE37704_featurecounts.csv", header=TRUE, row.names=1)
coldata <- read.csv("http://www.dursi.ca/content/images/GSE37704_metadata.csv", header=TRUE, row.names=1)

# Remove first column (length)
countdata <- countdata[-1]

# Convert to matrix
countdata <- as.matrix(countdata)
head(countdata)
```

Out[1]:

	SRR493366	SRR493367	SRR493368	SRR493369	SRR493370	SRR493371
ENSG00000186092	0	0	0	0	0	0

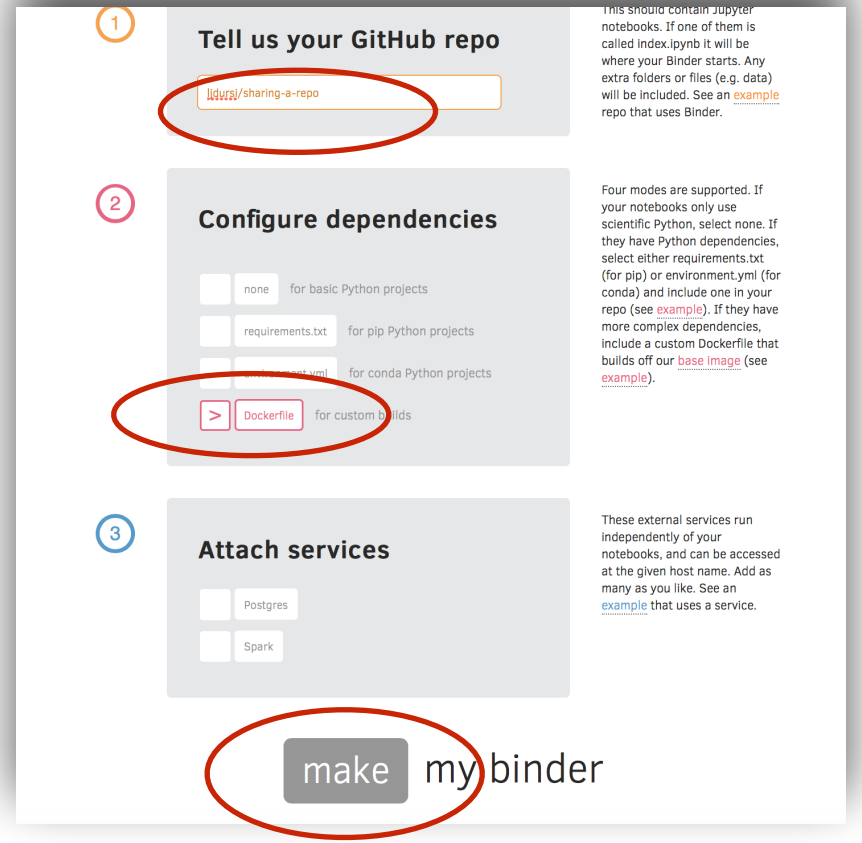
mybinder.org

- As before, upload some more files to your repo:
- Dockerfile
- install-irkernel.R
- install-other.R



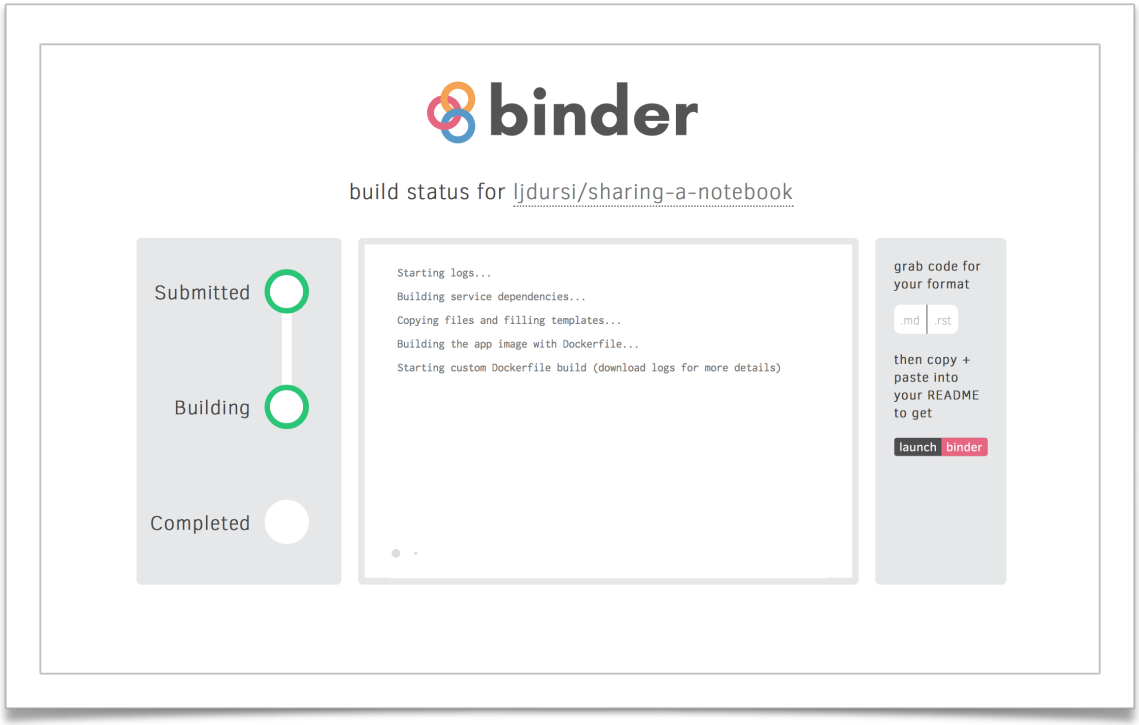
mybinder.org

- Then on mybinder, make a binder image
- Provide the repo (ljdursi/sharing-a-notebook),
- Using a Dockerfile
- Then hit make



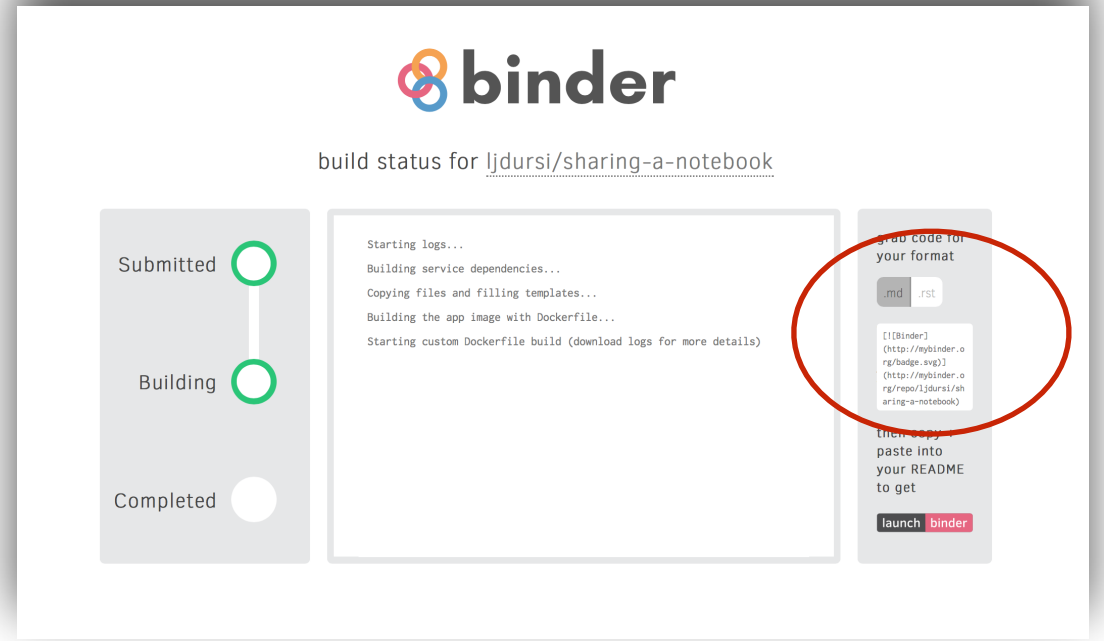
mybinder.org

- This will take several minutes.
- An image is being built just for your files, and then pushed out to the entire google cluster that my binder uses.



mybinder.org

- It will give you the option of copying some markdown to edit into your README.md
- This will give you the "launch binder" button as shown below



mybinder.org

- mybinder.org actually provides runnable/editable copies of your notebook
- An HHMI lab (The Freeman Lab) pays $$ for the compute time on Google
- Limits on sizes, compute time