Enabling Reproducible NGS Analysis Through Automated Jupyter Pipelines

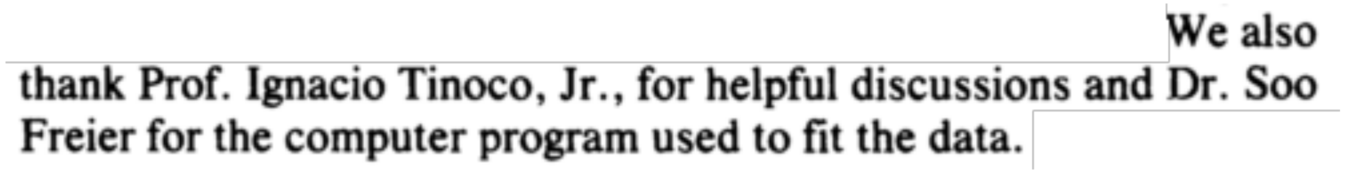
Amanda Birmingham

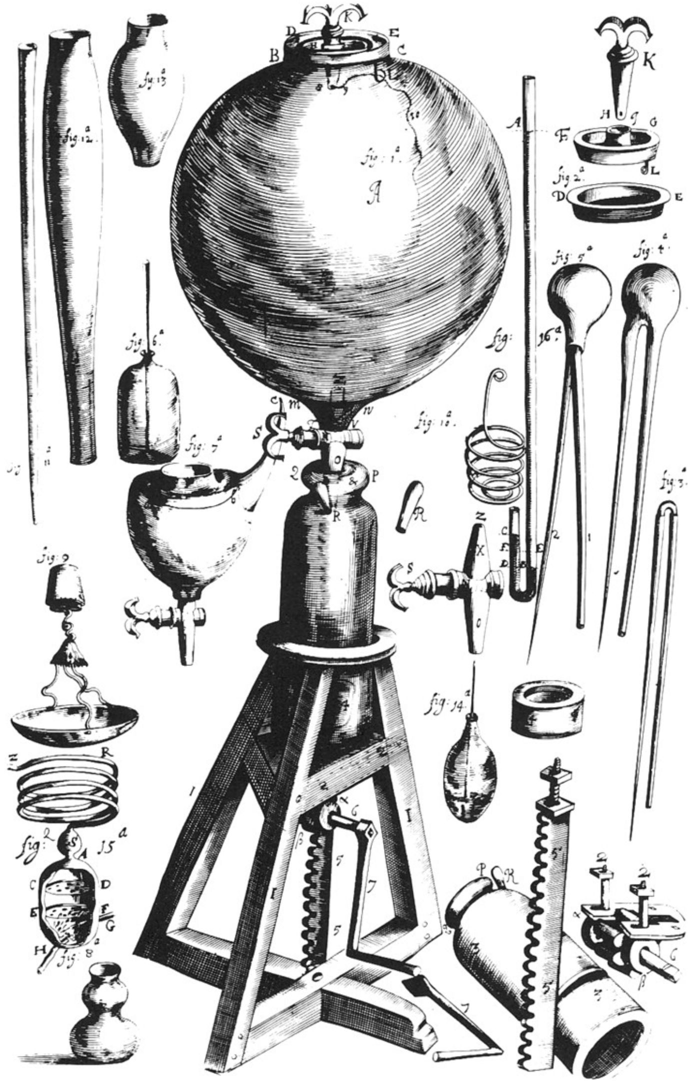
Senior Bioinformatics Engineer

Center for Computational Biology & Bioinformatics, UCSD

Reproducible Research

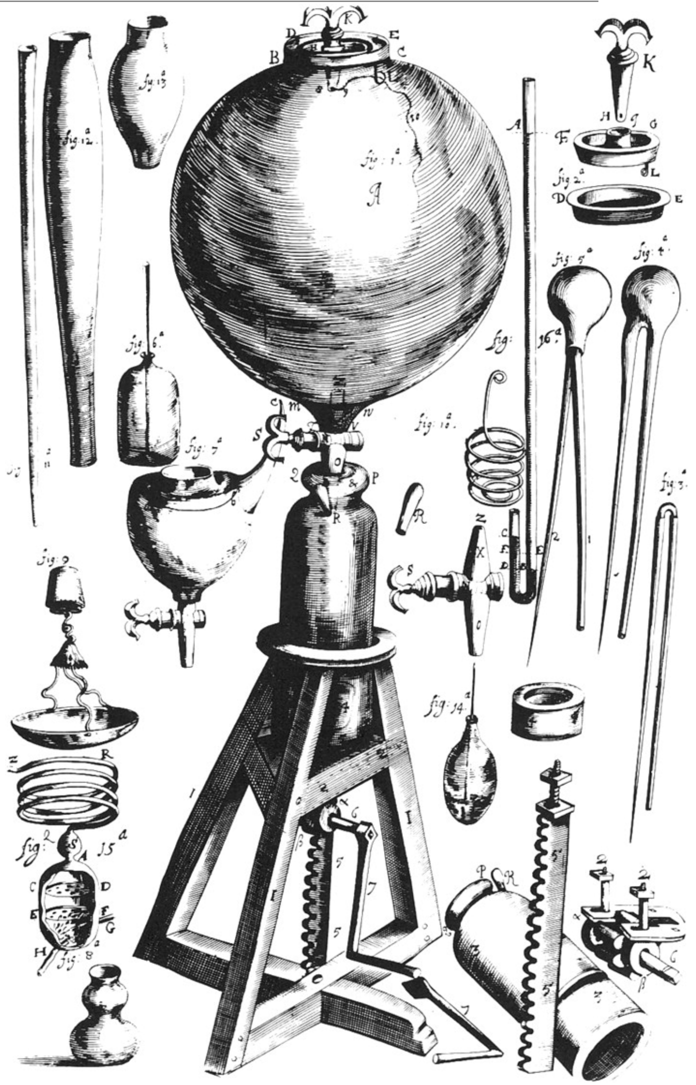
- Repeatability & reproducibility are key to the scientific method
  - In 1663, only Robert Boyle and Christiaan Huygens could produce a vacuum—and their findings didn't agree
- Informatics *should* be at the forefront of reproducible research
  - Doing the same thing over and over is what computers do best!
  - But has taken a long time for methods reports for computational work to become as good as those for wet lab work
  - Ex: Proc Natl Acad Sci USA. 1986 Jun;83(11):3746-50


We also thank Prof. Ignacio Tinoco, Jr., for helpful discussions and Dr. Soo Freier for the computer program used to fit the data.



Reproducible Research

- Repeatability & reproducibility are key to the scientific method
  - In 1663, only Robert Boyle and Christiaan Huygens could produce a vacuum—and their findings didn't agree
- Informatics *should* be at the forefront of reproducible research
  - Doing the same thing over and over is what computers do best!
  - But has taken a long time for methods reports for computational work to become as good as those for wet lab work
  - Ex: Proc Natl Acad Sci USA. 1986 Jun;83(11):3746-50
  - Progress:
    - "Alignments were run"
    - "Alignments were run with BLAST"
    - "Alignments were run with BLASTN version 2.2.6 against human"
    - "Alignments were run with NCBI BLASTN v.2.2.9 using the command `blastn -W 7 -q -1 -F F` against the NCBI RefSeq release 80 human transcriptome"
- Parity with wet-lab methods shouldn't be the end of the road!

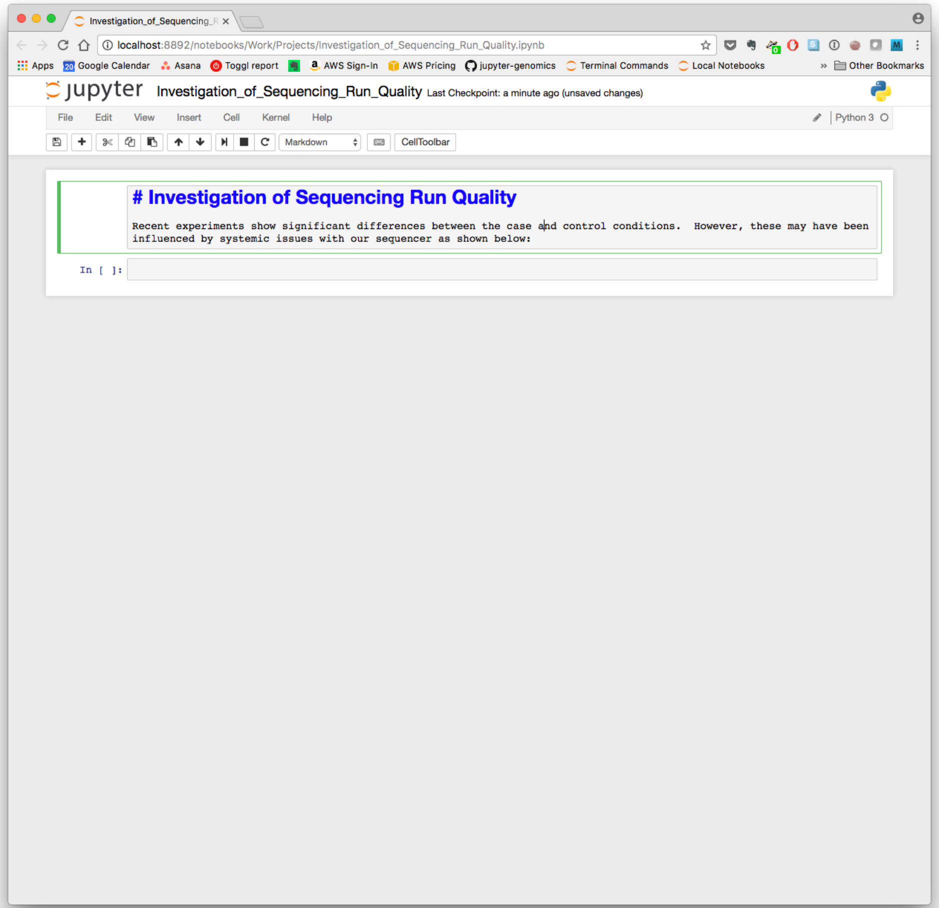


What Is Jupyter?

- What Is Jupyter?
  - "Open source, interactive data science and scientific computing across over 40 programming languages"
    - Grew out of the IPython project, which started in 2001 when Dr. Fernando Perez was procrastinating on his Physics PhD :)
  - A "literate computing" environment, "weaving of a narrative directly into a live computation, interleaving text with code and results to construct a complete piece" --Fernando Perez
- Computing platform is named "jupyter" because early languages were julia, python, and R
  - Community-maintained kernels for other languages: Bash, C, C++, C#, Fortran, Go, Haskell, Javascript, Lisp, Mathematica, Matlab, Perl, PHP, Powershell, Ruby, SAS, Scala, Scheme, and many more
- Most well-known for a web-based "notebook" system
  - Allows writing & running of code from browser environment
  - Can mix in HTML, links, images, interactive controls, extensions

Jupyter logo courtesy of http://jupyter.org/

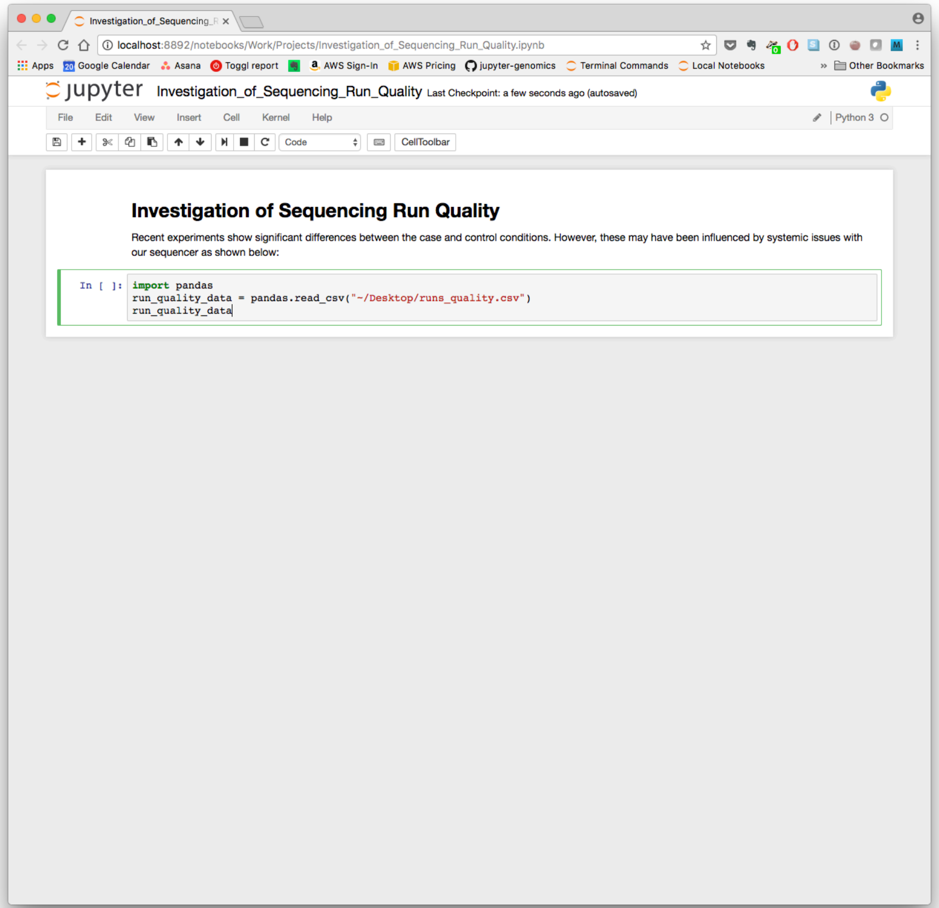
What *Is* Jupyter, Really?


Investigation_of_Sequencing_R
localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
Apps
Google Calendar
Asana
Toggl report
AWS Sign-In
AWS Pricing
jupyter-genomics
Terminal Commands
Local Notebooks
Other Bookmarks
jupyter
Investigation_of_Sequencing_Run_Quality
Last Checkpoint: a minute ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Markdown
CellToolbar
In [ ]:

```
# Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions.  However, these may have been
influenced by systemic issues with our sequencer as shown below:
```

What *Is* Jupyter, Really?


Investigation_of_Sequencing_R
localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
Apps
Google Calendar
Asana
Toggl report
AWS Sign-In
AWS Pricing
jupyter-genomics
Terminal Commands
Local Notebooks
Other Bookmarks
jupyter
Investigation_of_Sequencing_Run_Quality Last Checkpoint: a few seconds ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Code
CellToolbar

Investigation of Sequencing Run Quality

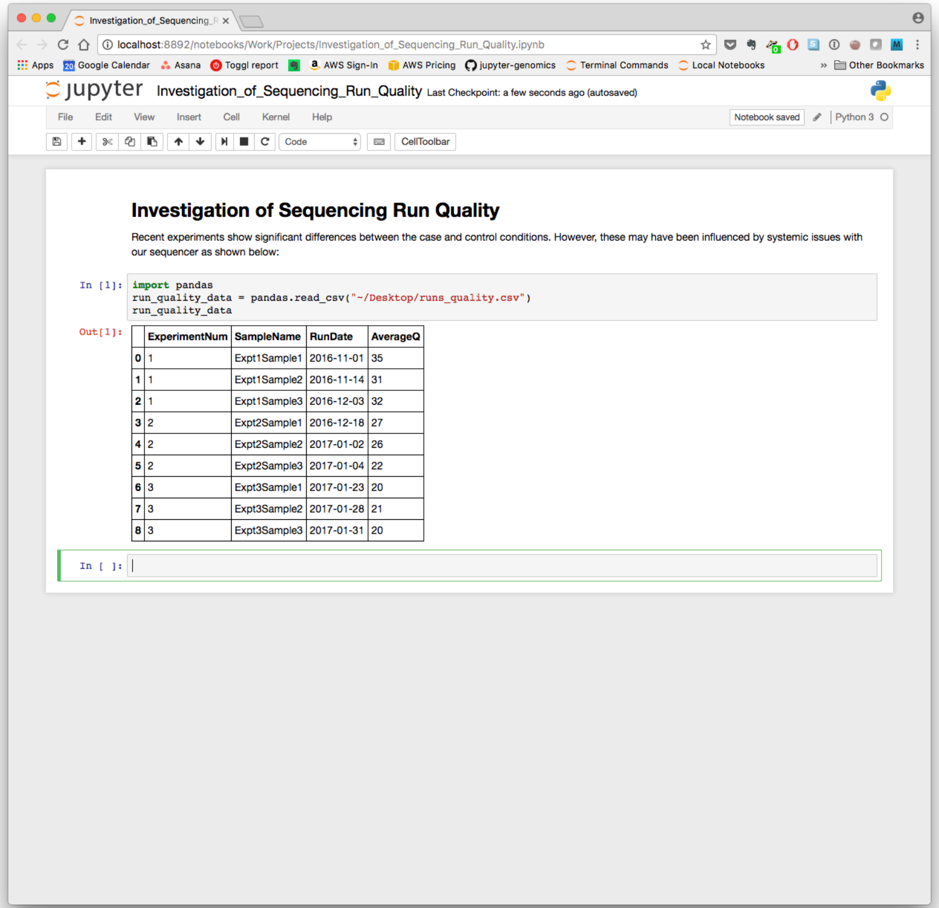
Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

```
In [ ]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

UC San Diego
SCHOOL OF MEDICINE

CCBB CENTER FOR COMPUTATIONAL BIOLOGY & BIOINFORMATICS

What *Is* Jupyter, Really?


Investigation_of_Sequencing_R
localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
Apps
Google Calendar
Asana
Toggl report
AWS Sign-In
AWS Pricing
jupyter-genomics
Terminal Commands
Local Notebooks
Other Bookmarks
jupyter
Investigation_of_Sequencing_Run_Quality Last Checkpoint: a few seconds ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Notebook saved
Python 3
Code
CellToolbar

Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

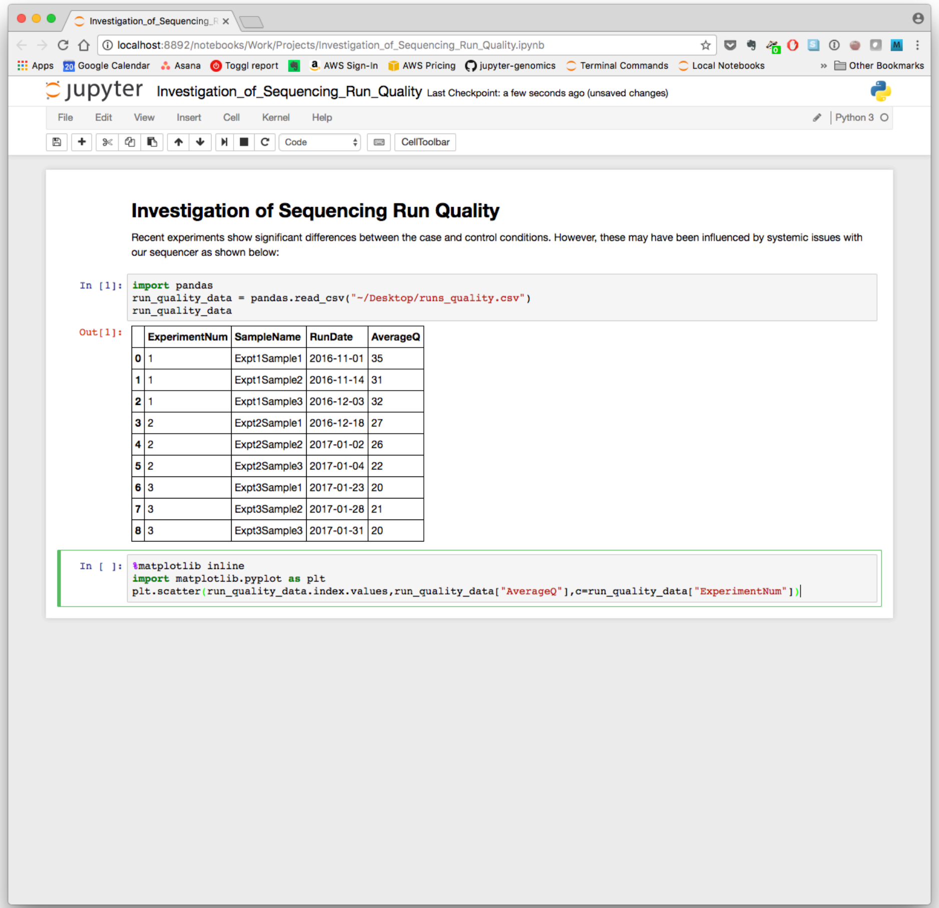
```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

Out[1]:

	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

```
In [ ]:
```

What *Is* Jupyter, Really?


Investigation_of_Sequencing_R
localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
Apps
Google Calendar
Asana
Toggl report
AWS Sign-In
AWS Pricing
jupyter-genomics
Terminal Commands
Local Notebooks
Other Bookmarks
jupyter
Investigation_of_Sequencing_Run_Quality Last Checkpoint: a few seconds ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Code
CellToolbar

Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

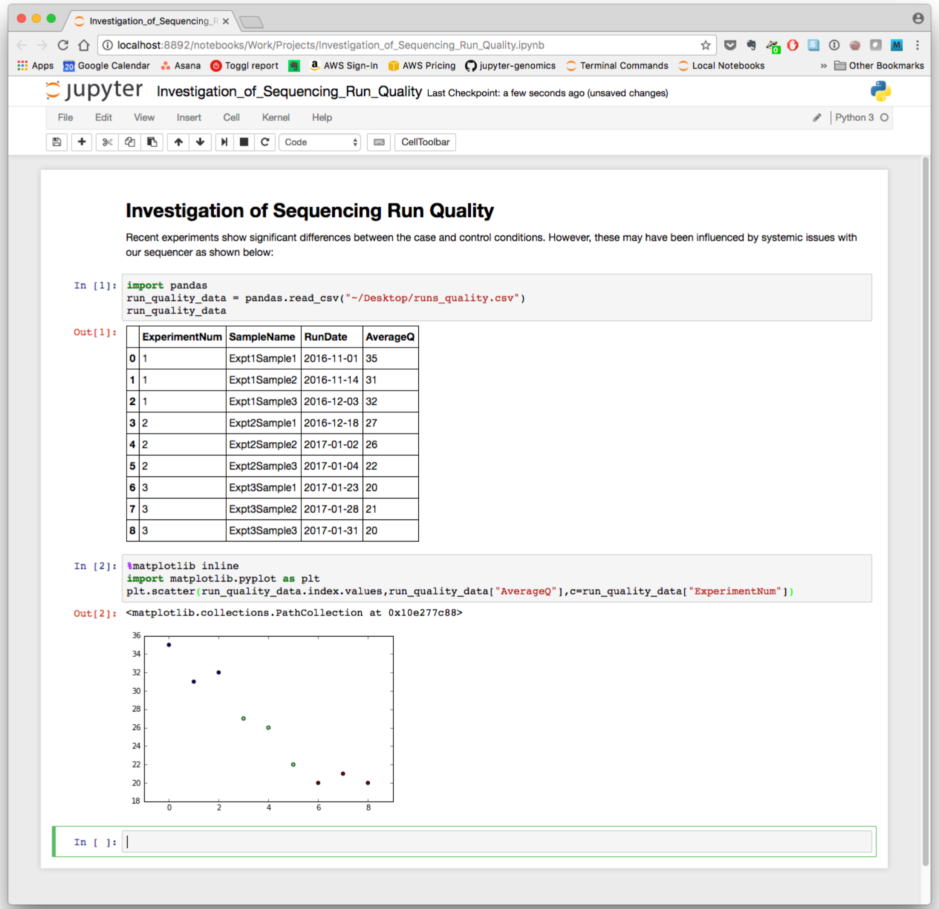
```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

Out[1]:

	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

```
In [ ]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.scatter(run_quality_data.index.values,run_quality_data["AverageQ"],c=run_quality_data["ExperimentNum"])
```

What *Is* Jupyter, Really?


Investigation_of_Sequencing_R
localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
Apps
Google Calendar
Asana
Toggl report
AWS Sign-In
AWS Pricing
jupyter-genomics
Terminal Commands
Local Notebooks
Other Bookmarks
jupyter
Investigation_of_Sequencing_Run_Quality
Last Checkpoint: a few seconds ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Code
CellToolbar

Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

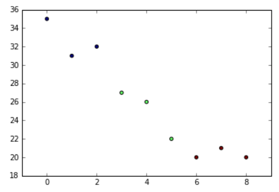
```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

Out[1]:

	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

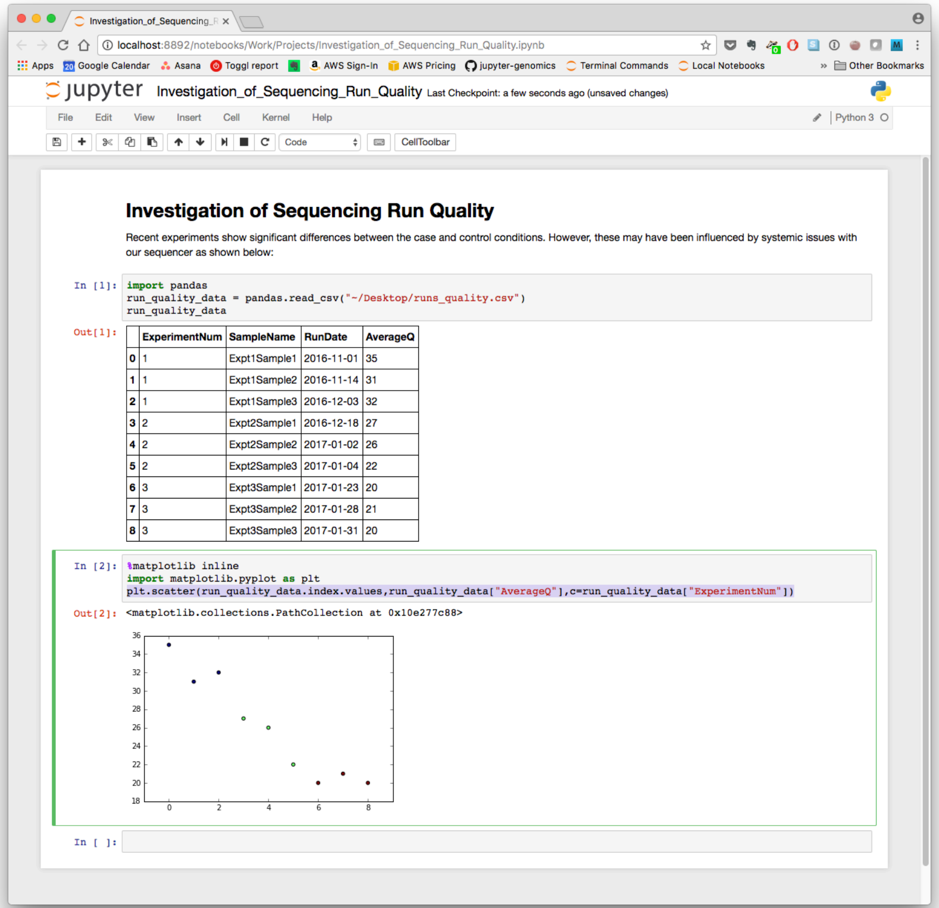
```
In [2]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.scatter(run_quality_data.index.values,run_quality_data["AverageQ"],c=run_quality_data["ExperimentNum"])

Out[2]: <matplotlib.collections.PathCollection at 0x10e277c88>
```



```
In [ ]:
```

What *Is* Jupyter, Really?



Investigation_of_Sequencing_Run_Quality Last Checkpoint: a few seconds ago (unsaved changes)

Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

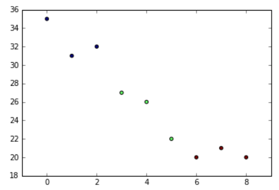
```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

Out[1]:

	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

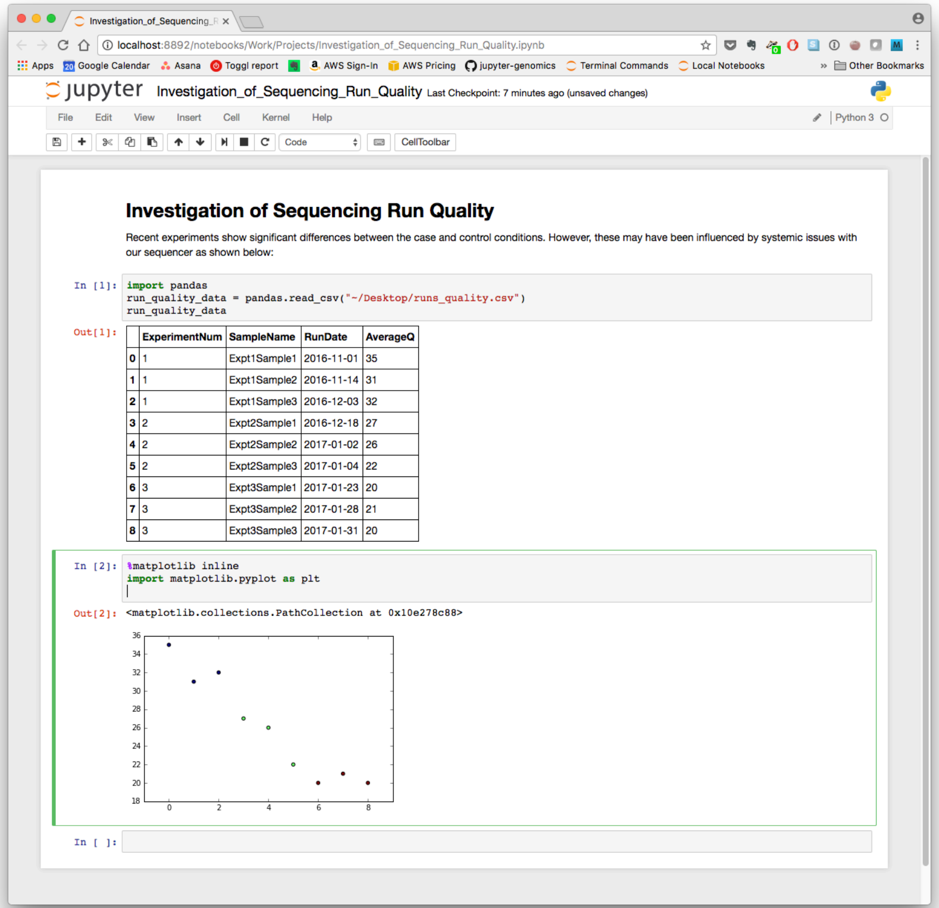
```
In [2]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.scatter(run_quality_data.index.values,run_quality_data["AverageQ"],c=run_quality_data["ExperimentNum"])

Out[2]: <matplotlib.collections.PathCollection at 0x10e277c88>
```



```
In [ ]:
```

What *Is* Jupyter, Really?


Investigation_of_Sequencing_R
localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
Apps
Google Calendar
Asana
Toggl report
AWS Sign-In
AWS Pricing
jupyter-genomics
Terminal Commands
Local Notebooks
Other Bookmarks
jupyter
Investigation_of_Sequencing_Run_Quality Last Checkpoint: 7 minutes ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Code
CellToolbar

Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

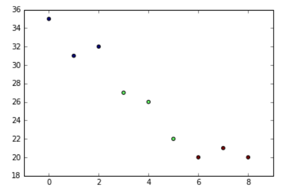
```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

Out[1]:

	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

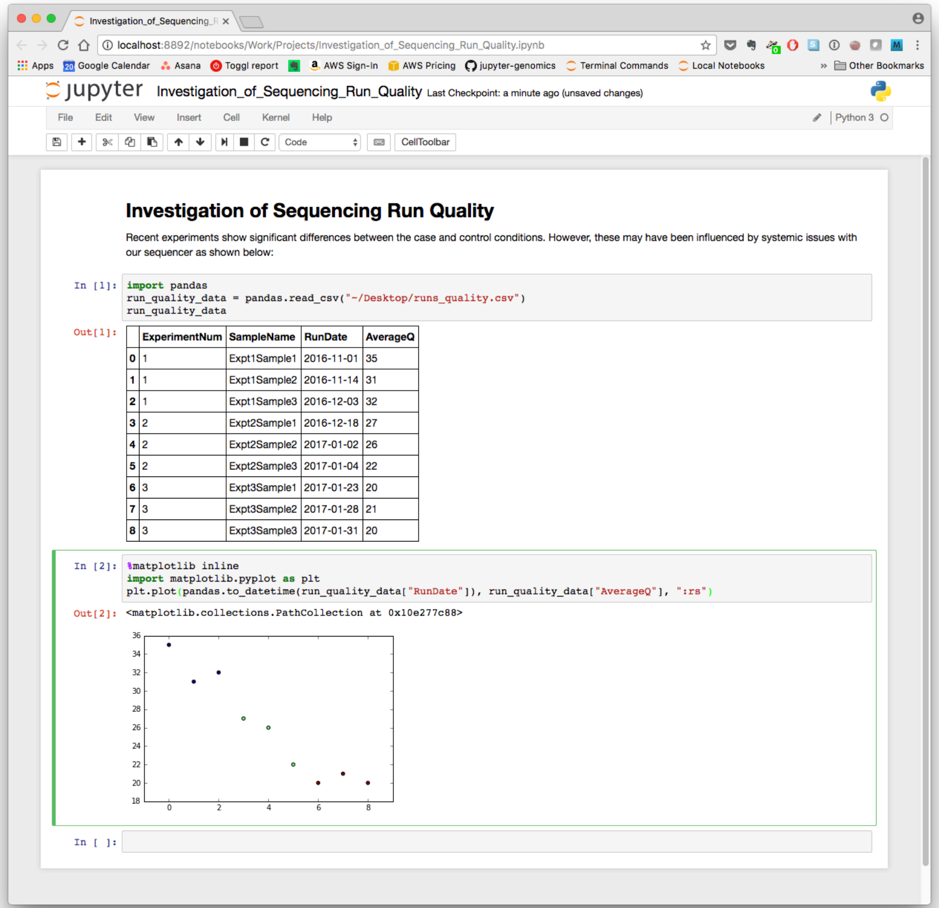
```
In [2]: %matplotlib inline
        import matplotlib.pyplot as plt

Out[2]: <matplotlib.collections.PathCollection at 0x10e278c88>
```



```
In [ ]:
```

What *Is* Jupyter, Really?


localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
jupyter Investigation_of_Sequencing_Run_Quality Last Checkpoint: a minute ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Python 3
Code
CellToolbar

Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

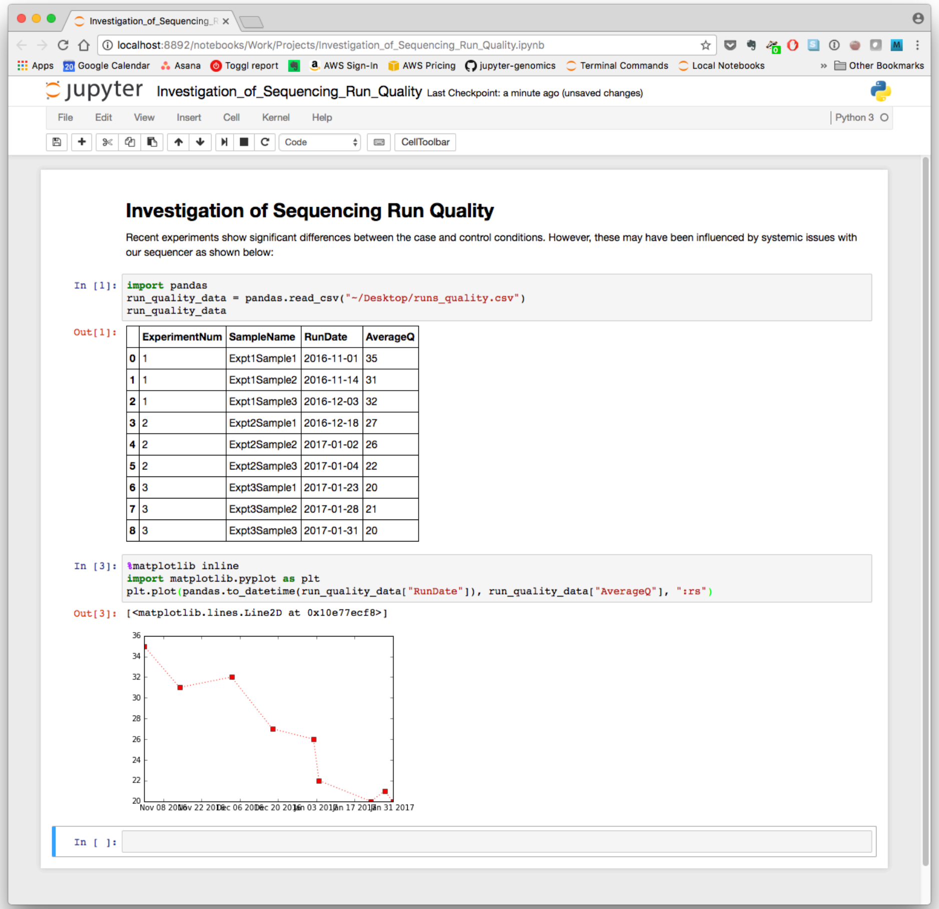
Out[1]:

	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

```
In [2]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.plot(pandas.to_datetime(run_quality_data["RunDate"]), run_quality_data["AverageQ"], ":rs")
Out[2]: <matplotlib.collections.PathCollection at 0x10e277c88>
```

```
In [ ]:
```

What *Is* Jupyter, Really?


Investigation_of_Sequencing_R
localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
Apps
Google Calendar
Asana
Toggl report
AWS Sign-In
AWS Pricing
jupyter-genomics
Terminal Commands
Local Notebooks
Other Bookmarks
jupyter
Investigation_of_Sequencing_Run_Quality
Last Checkpoint: a minute ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Code
CellToolbar

Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

Out[1]:

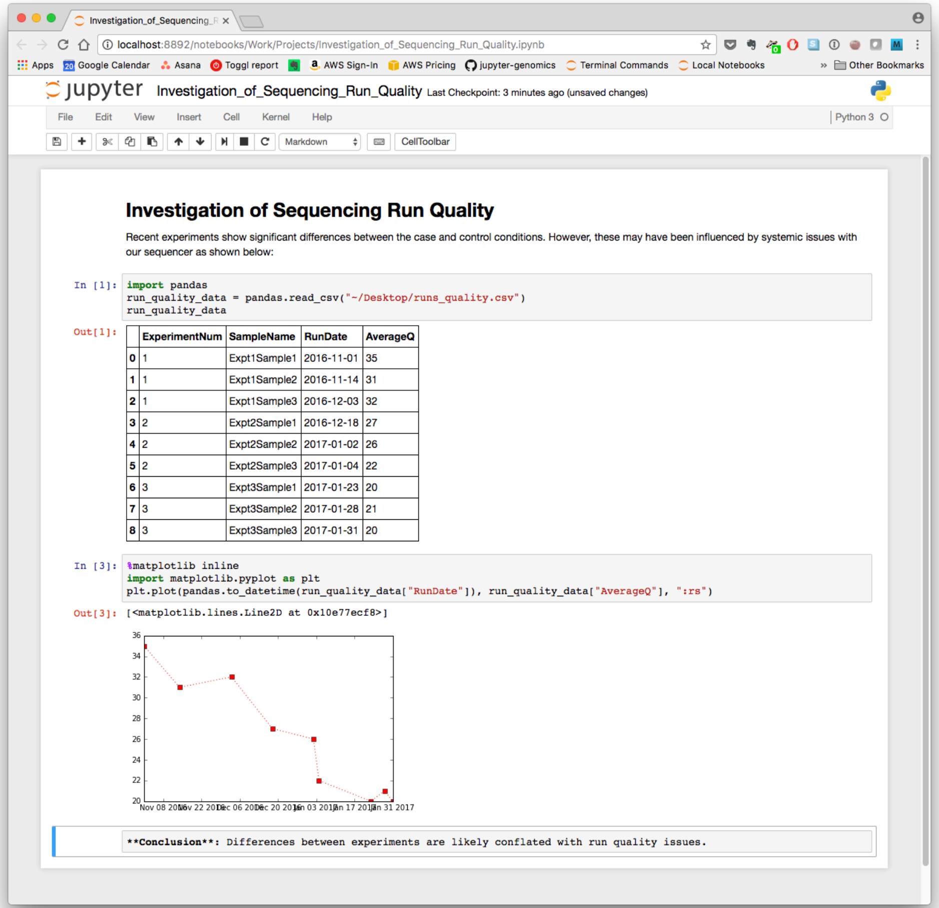
	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

```
In [3]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.plot(pandas.to_datetime(run_quality_data["RunDate"]), run_quality_data["AverageQ"], ":rs")

Out[3]: [<matplotlib.lines.Line2D at 0x10e77ecf8>]
```

```
In [ ]:
```

What *Is* Jupyter, Really?



Investigation of Sequencing Run Quality

Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

Out[1]:

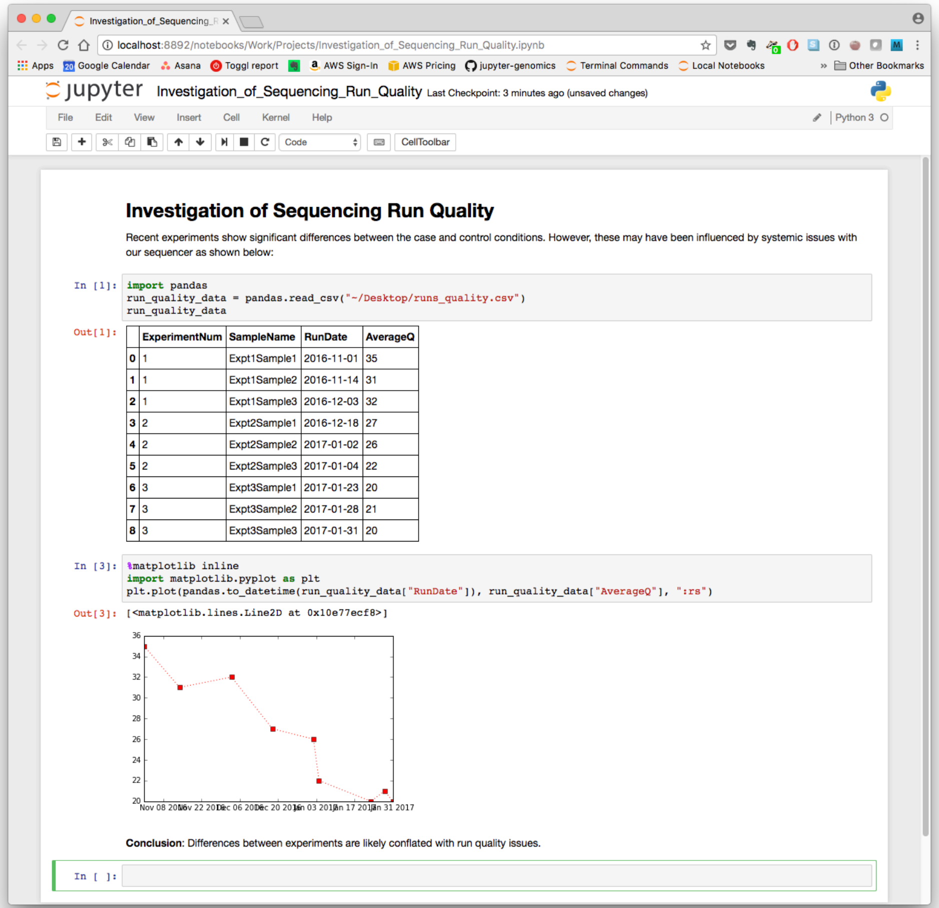
	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

```
In [3]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.plot(pandas.to_datetime(run_quality_data["RunDate"]), run_quality_data["AverageQ"], ":rs")

Out[3]: [<matplotlib.lines.Line2D at 0x10e77ecf8>]
```

```
**Conclusion**: Differences between experiments are likely conflated with run quality issues.
```

What *Is* Jupyter, Really?


Investigation_of_Sequencing_R
localhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.ipynb
Apps
Google Calendar
Asana
Toggl report
AWS Sign-In
AWS Pricing
jupyter-genomics
Terminal Commands
Local Notebooks
Other Bookmarks
jupyter
Investigation_of_Sequencing_Run_Quality Last Checkpoint: 3 minutes ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Code
CellToolbar

Investigation of Sequencing Run Quality

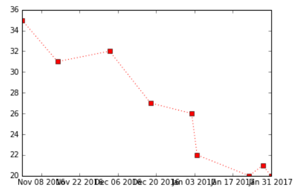
Recent experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with our sequencer as shown below:

```
In [1]: import pandas
        run_quality_data = pandas.read_csv("~/Desktop/runs_quality.csv")
        run_quality_data
```

Out[1]:

	ExperimentNum	SampleName	RunDate	AverageQ
0	1	Expt1Sample1	2016-11-01	35
1	1	Expt1Sample2	2016-11-14	31
2	1	Expt1Sample3	2016-12-03	32
3	2	Expt2Sample1	2016-12-18	27
4	2	Expt2Sample2	2017-01-02	26
5	2	Expt2Sample3	2017-01-04	22
6	3	Expt3Sample1	2017-01-23	20
7	3	Expt3Sample2	2017-01-28	21
8	3	Expt3Sample3	2017-01-31	20

```
In [3]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.plot(pandas.to_datetime(run_quality_data["RunDate"]), run_quality_data["AverageQ"], ":rs")
Out[3]: [<matplotlib.lines.Line2D at 0x10e77ecf8>]
```

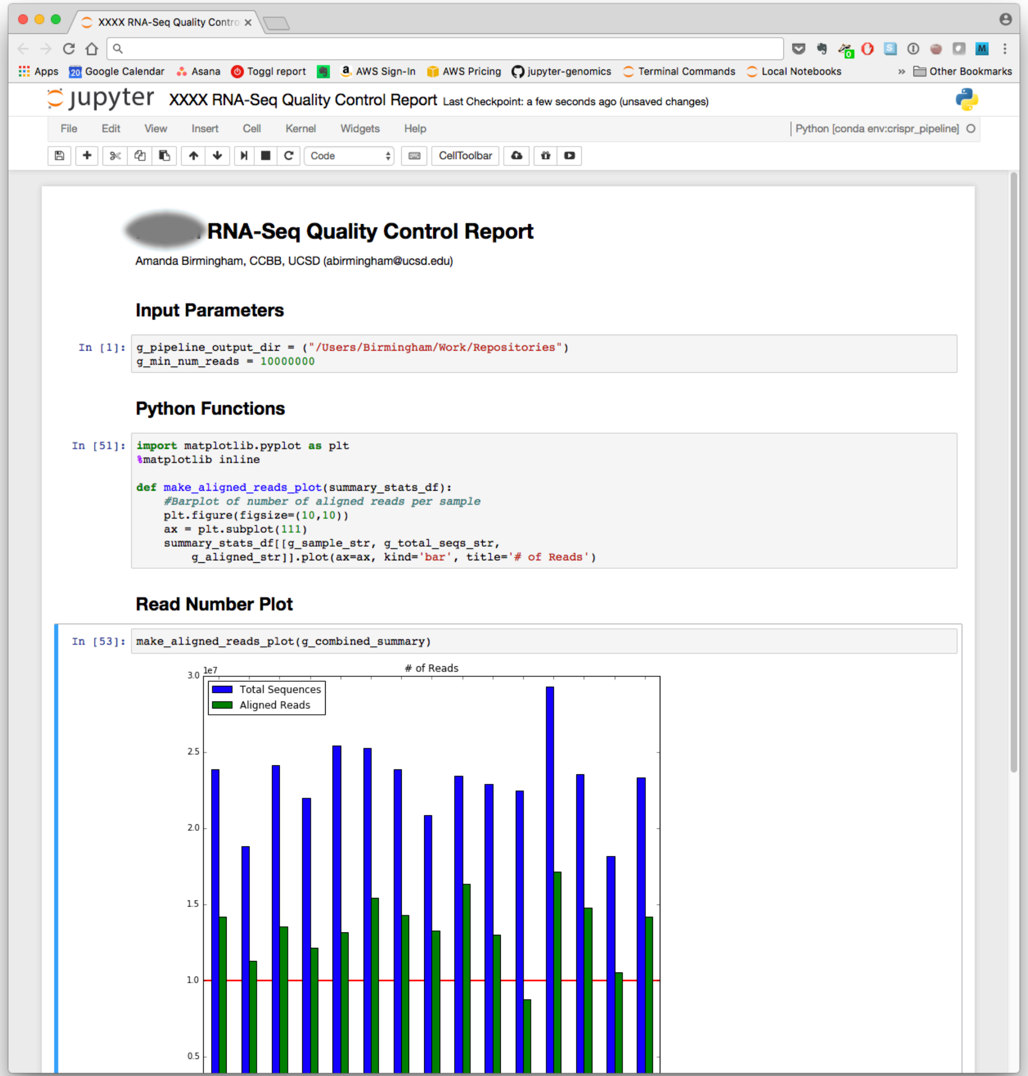


Conclusion: Differences between experiments are likely conflated with run quality issues.

```
In [ ]:
```

Jupyter Notebooks: Friend or Foe?

- Are notebooks the key to reproducibility?
  - Data Carpentry offers an entire workshop on "Reproducible Research using Jupyter Notebooks”
- Easy to save, modify, and extend
  - Great for rerunning or tweaking previous data analyses
- CCBB delivers analyses as notebooks
  - Report becomes more than a record—it is itself a tool!
- Notebooks’ *greatest strength* is interactivity
  - Between input and output
  - Between (e.g.) Python and R
  - Between narrative and code
  - Between material and reader



[redacted] RNA-Seq Quality Control Report

Amanda Birmingham, CCBB, UCSD (abirmingham@ucsd.edu)

Input Parameters

```
In [1]: g_pipeline_output_dir = ("/Users/Birmingham/Work/Repositories")
        g_min_num_reads = 10000000
```

Python Functions

```
In [51]: import matplotlib.pyplot as plt
         %matplotlib inline

         def make_aligned_reads_plot(summary_stats_df):
             #Barplot of number of aligned reads per sample
             plt.figure(figsize=(10,10))
             ax = plt.subplot(111)
             summary_stats_df[[g_sample_str, g_total_seqs_str,
                 g_aligned_str]].plot(ax=ax, kind='bar', title='# of Reads')
```

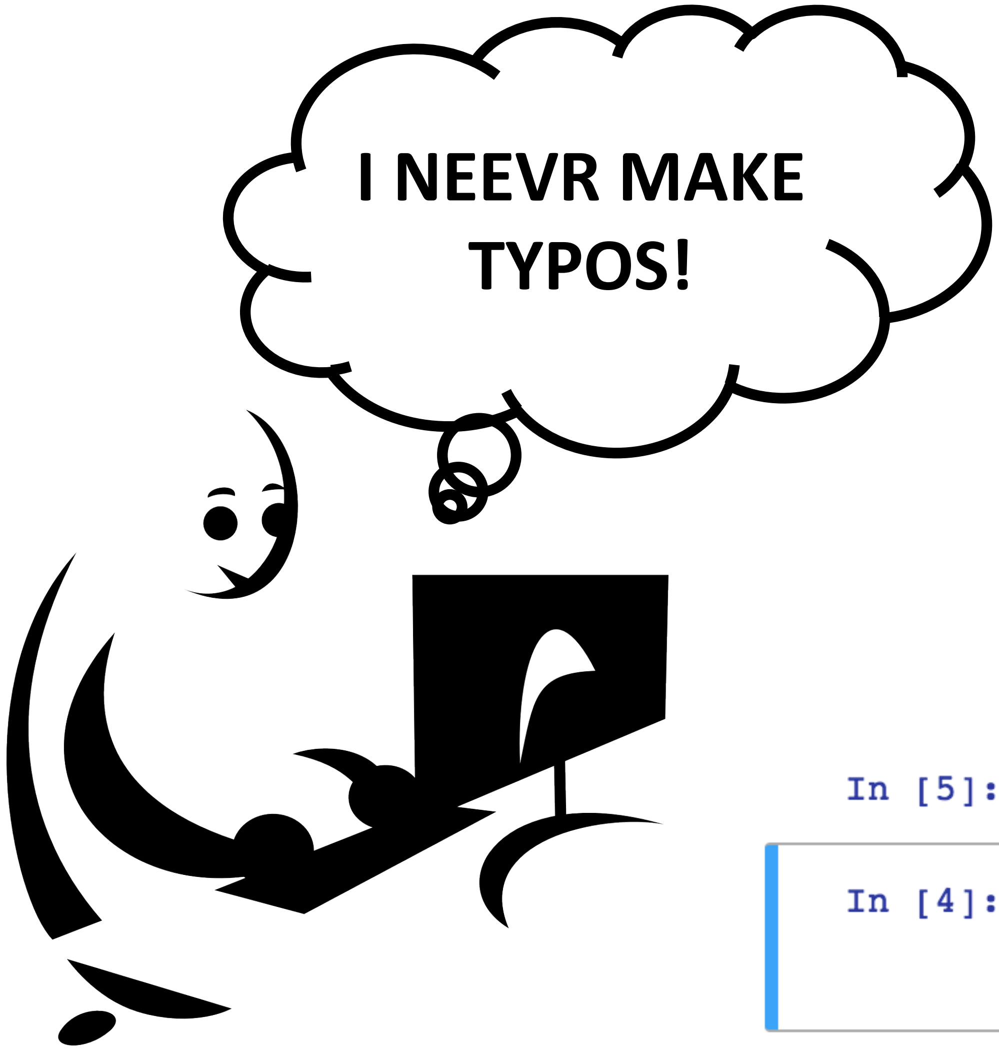
Read Number Plot

```
In [53]: make_aligned_reads_plot(g_combined_summary)
```

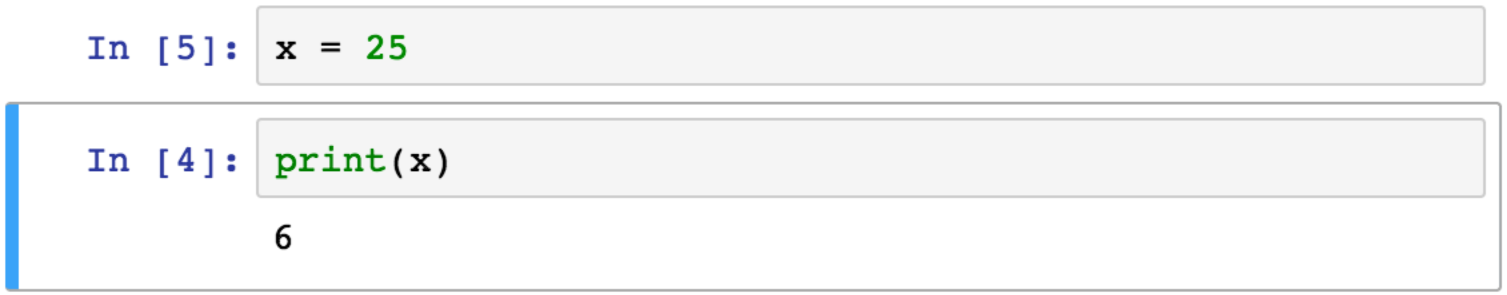
(Inter-)Actively Dangerous

- Interactivity can also be a *huge danger* to reproducibility
- Humans are inconsistent
  - We make unpredictable mistakes
  - Thus, “interactive”=“bad” for repetitive tasks
    - Like primary NGS analysis pipelines
- Jupyter Notebooks can be inconsistent, too
  - Changing code/variables in a notebook does NOT rerun cells that depend on that change
  - In fact, doesn’t even clear old outputs!

(Inter-)Actively Dangerous



- Interactivity can also be a *huge danger* to reproducibility
- Humans are inconsistent
  - We make unpredictable mistakes
  - Thus, "interactive"="bad" for repetitive tasks
    - Like primary NGS analysis pipelines
- Jupyter Notebooks can be inconsistent, too
  - Changing code/variables in a notebook does NOT rerun cells that depend on that change
  - In fact, doesn't even clear old outputs!



```
In [5]: x = 25

In [4]: print(x)

        6
```

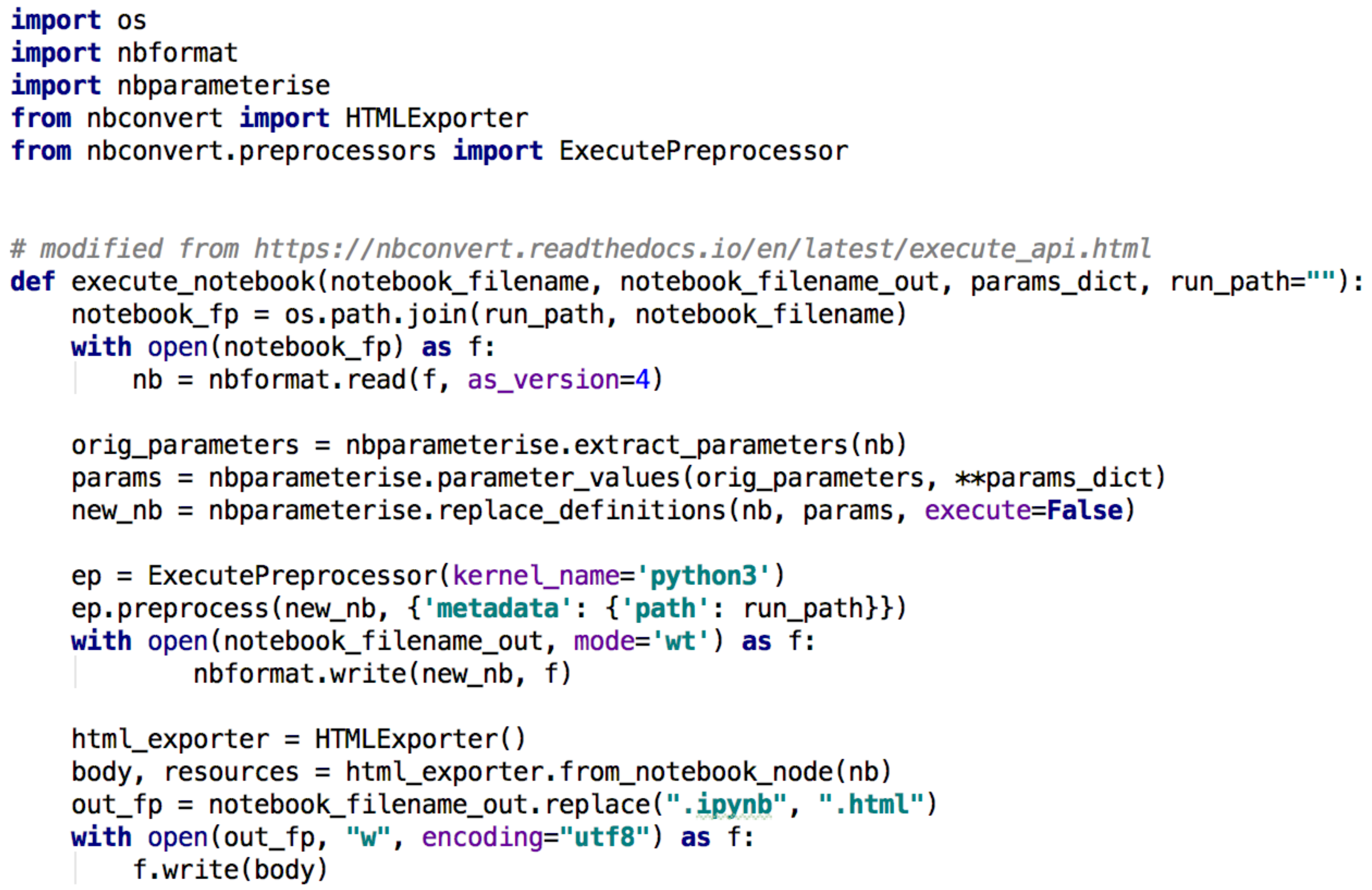
(Inter-)Actively Dangerous

- Interactivity can also be a *huge danger* to reproducibility
- Humans are inconsistent
  - We make unpredictable mistakes
  - Thus, “interactive”=“bad” for repetitive tasks
    - Like primary NGS analysis pipelines
- Jupyter Notebooks can be inconsistent, too
  - Changing code/variables in a notebook does NOT rerun cells that depend on that change
  - In fact, doesn’t even clear old outputs!
  - Thus, “interactive”=“bad” for important records
    - Like experimental records (i.e., methods)
- Do we have to give up other advantages of Jupyter Notebooks when building pipelines and recording methods?

Scripting Jupyter Notebooks

- No! We can have our cake and eat it, too ☺
- Jupyter ships with **nbconvert** package that can read, write, and execute notebooks from Python
- An extension, **nbparameterise** (note British spelling) allows injection of new variable values
- **nbconvert** and **nbformat** (also built-in) can output notebooks and static html, respectively
- With these three pieces, we can script pipelines built from Jupyter Notebooks
  - Notebooks give readability and reusability
  - Script prevents human errors and speeds execution
  - HTML output of notebooks provides read-only record of methods
- Entire approach takes less than one page of code

Scripting Jupyter Notebooks



```python
import os
import nbformat
import nbparameterise
from nbconvert import HTMLExporter
from nbconvert.preprocessors import ExecutePreprocessor


# modified from https://nbconvert.readthedocs.io/en/latest/execute_api.html
def execute_notebook(notebook_filename, notebook_filename_out, params_dict, run_path=""):
    notebook_fp = os.path.join(run_path, notebook_filename)
    with open(notebook_fp) as f:
        nb = nbformat.read(f, as_version=4)

    orig_parameters = nbparameterise.extract_parameters(nb)
    params = nbparameterise.parameter_values(orig_parameters, **params_dict)
    new_nb = nbparameterise.replace_definitions(nb, params, execute=False)

    ep = ExecutePreprocessor(kernel_name='python3')
    ep.preprocess(new_nb, {'metadata': {'path': run_path}})
    with open(notebook_filename_out, mode='wt') as f:
            nbformat.write(new_nb, f)

    html_exporter = HTMLExporter()
    body, resources = html_exporter.from_notebook_node(nb)
    out_fp = notebook_filename_out.replace(".ipynb", ".html")
    with open(out_fp, "w", encoding="utf8") as f:
        f.write(body)
```

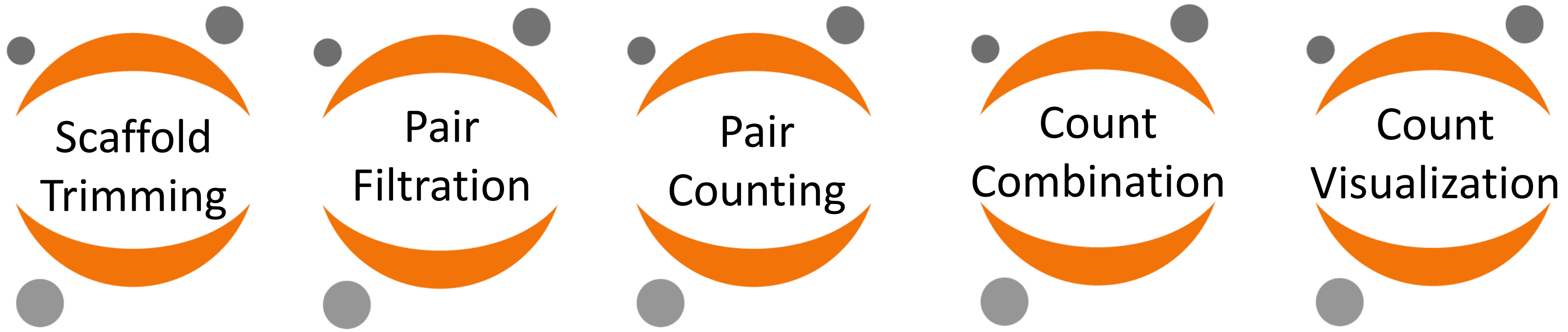
Scripting Jupyter Notebooks

- No! We can have our cake and eat it, too ☺
- Jupyter ships with **nbconvert** package that can read, write, and execute notebooks from Python
- An extension, **nbparameterise** (note British spelling) allows injection of new variable values
- **nbconvert** and **nbformat** (also built-in) can output notebooks and static html, respectively
- With these three pieces, we can script pipelines built from Jupyter Notebooks
  - Notebooks give readability and reusability
  - Script prevents human errors and speeds execution
  - HTML output of notebooks provides read-only record of methods
- Entire approach takes less than one page of code

```
execute_notebook("mynotebook.ipynb", "my_new_notebook.ipynb", {"x": 6, "y": "blue"})
```

Notebooks in the Wild

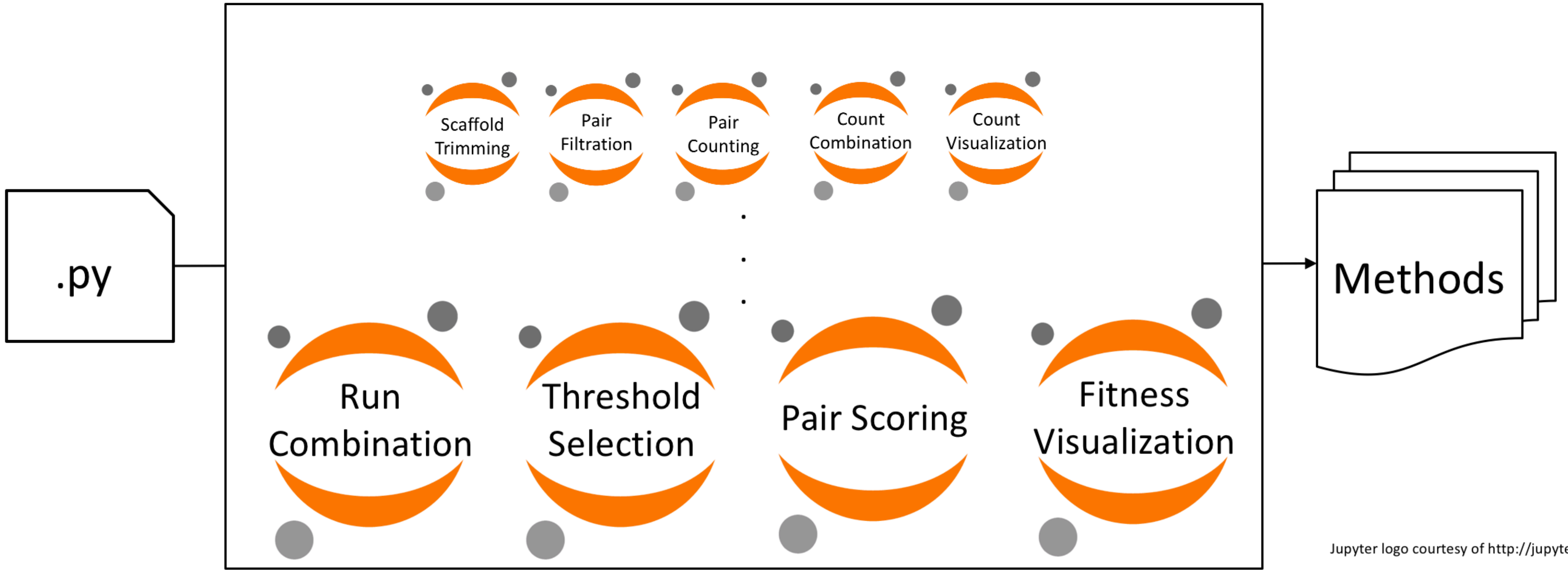
- A sample NGS pipeline using Jupyter Notebooks
  - Goal: identify gene pairs with synergistic survival effects (positive or negative)
  - Experimental system:: Dual-gene knock-outs in human cell lines using CRISPR
  - Read-out: number of instances of each CRISPR guide in final population, assessed by NGS



Jupyter logo courtesy of http://jupyter.org/

Notebooks in the Wild

- A sample NGS pipeline using Jupyter Notebooks
  - Goal: identify gene pairs with synergistic survival effects (positive or negative)
  - Experimental system:: Dual-gene knock-outs in human cell lines using CRISPR
  - Read-out: number of instances of each CRISPR guide in final population, assessed by NGS



Jupyter logo courtesy of http://jupyter.org/

Conclusions

- Jupyter Notebooks are a fantastic tool for data analysis—*but*:
- Their twin goals of interactivity and reproducibility are often at odds
- Notebooks can be scripted to reduce error potential
  - And notebook-based pipelines self-document nicely!
- CCBB has implemented a sample Jupyter-based pipeline for NGS data from dual CRISPR screens
  - Pipeline is part of work with Dr.s Prashant Mali & Trey Ideker, now in press at Nature Methods
  - Code is available in the "CRISPR" section of CCBB's jupyter-genomics repository on GitHub
    - https://github.com/ucsd-ccbb/jupyter-genomics
- CCBB's Data Science Blog gives a further intro to notebook scripting
  - http://ccbb.bio/outreach/data-science-blog/
- Reproducible data analysis is hard work—but worth the effort!

http://ccbb.bio

Acknowledgments

- Fernando Perez & the Jupyter Project!
- Dual CRISPR Team
  - Mali lab
  - Ideker lab
- CCBB Team
  - Katie Fisch (Director)
  - Roman Sasik
  - Guorong Xu
  - Brin Rosenthal
- Our funders
  - UC San Diego Health Sciences
  - CTRI Center for Accelerating Drug Development (CADD) – Grant UL1TR001442

UC San Diego HEALTH SCIENCES

UC San Diego Altman Clinical and Translational Research Institute