Enabling Reproducible NGS Analysis Through Automated Jupyter Pipelines

Amanda Birmingham

Senior Bioinformatics Engineer

Center for Computational Biology & Bioinformatics, UCSD



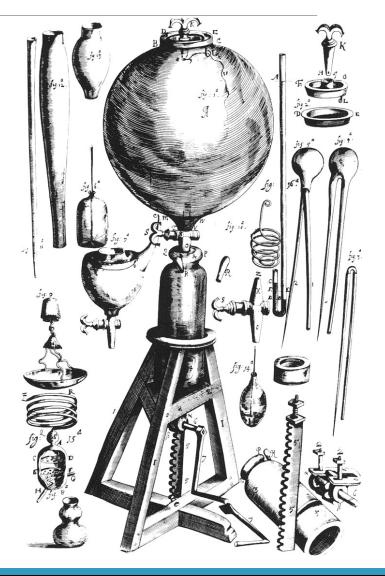


Reproducible Research

- 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.

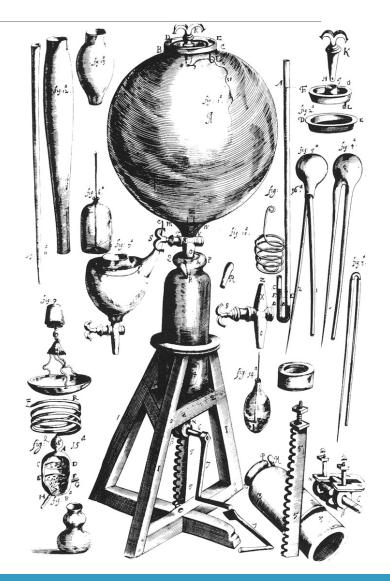






Reproducible Research

- 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







Investigation of Sequencing. I x	Θ
C 🛆 🛈 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 Bookr	marks
CJUPYTEY Investigation_of_Sequencing_Run_Quality Last Checkpoint: a minute ago (unsaved changes)	
File Edit View Insert Cell Kernel Help	
E + ≫ 42 E + ₩ E C Markdown : E CellToolbar	
# Investigation of Permanaing Due Quality	
# 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 []:	





Investiga	tion.of.Sequencing.F ×	θ
	calhost:8892/notebooks/Work/Projects/Investigation_of_Sequencing_Run_Quality.jpynb 🖈 🛡 🏘 🔏 🚺	S 0 👄 🗖 M E
	endar 🔥 Asana 🧿 Toggl report 📕 🕭 AWS Sign-in 🔋 AWS Pricing 🎧 jupyter-genomics 🔿 Terminal Commands 🔿 Local Notebooks	» 🗎 Other Bookmarks
🗢 jupyte	CT Investigation_of_Sequencing_Run_Quality Last Checkpoint: a few seconds ago (autosaved)	
File Edit	View Insert Cell Kernel Help	Python 3 O
B + % (2 The A V N The C 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 our sequencer as shown below:	issues with
In []:	<pre>import pandas run_quality_data = pandas.read_csv("-/Desktop/runs_quality.csv") run_quality_data</pre>	





		books/Work/Projec			ricing O jupyter-genomics	Terminal Commands		I IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
jupyte	er Investiga	tion_of_Seque	ncing_Rur	n_Quality	ast Checkpoint: a few seconds	ago (autosaved)		ć
File Edit	View Insert	Cell Kernel	Help				Notebook saved	Python 3
a + x	8 B 🛧 🖌	H C Cod	e \$	CellT	lbar			
	-			-	Quality se and control conditions. Ho	wever, these may have	been influenced by system	ic issues with
In [1]:	<pre>import pandas run_guality_d run_guality_d</pre>	ata = pandas.r	ead_csv("~/	'Desktop/1	ns_quality.csv")			
Out[1]:	ExperimentN	um 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		26				
	5 2	Expt2Sample3		22				
	6 3	Expt3Sample1		20				
	7 3	Expt3Sample2		21				
	8 3	Expt3Sample3	2017-01-31	20				
In []:	1							





🕨 🤇 😄 Investiga	tion	_of_Sequencing_R ×					
C 🗘 🛈 lo	calh	10st:8892/notebor	oks/Work/Projec	ts/Investigati	ion_of_Sequ	cing_Run_Quality.ipynb 🖈 🕻	7 🤊 🚜 () 💷 🛈 🗉 🗖
s 🙍 Google Cal	enda	ar 🔥 Asana 🄞 T	Toggl report 🛛 📃	a AWS Sign	-In 🧻 AWS	icing 🎧 jupyter-genomics 🗢 Terminal Commands 🗢 Local No	
ڬ jupyte	er	Investigatio	n_of_Seque	ncing_Rur	n_Quality	ist Checkpoint: a few seconds ago (unsaved changes)	2
File Edit	Vi	ew Insert C	Cell Kernel	Help			Python 3 O
8 + % (20	1□ ↑ ↓ H	C Code	¢ 6	CellT	bar	
	Re	nvestigatio	show significant	-	-	Quality e and control conditions. However, these may have been influer	nced by systemic issues with
In [1]:	ru	mport pandas in_quality_data in_quality_data		ad_csv("~/	Desktop/r	s_quality.csv")	
Out[1]:	Γ	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 []:	in	natplotlib inli nport matplotli t.scatter(run_	ib.pyplot as	plt .index.val	lues,run_q	lity_data["AverageQ"],c=run_quality_data["Exper	imentNum"])





🖲 🔍 🔵 📿 Investiga	ation_of_Seq	uencing_R >										(
	calhost:88	92/notebo	oks/Work/Proje	cts/Investigat	ion_of_Seq	quencing_Run	_Quality.ipynl	b		☆ 🖸 🕯	• ~ 0	5 O 👄 🗖 M
Apps 🙍 Google Cal	endar 🔒	Asana 🎯	Toggi report 🛛 🦷	a. AWS Sign	i-In 🧻 AW	VS Pricing O	jupyter-genon	nics 😄 Terminal	Commands			» 🗎 Other Bookmar
💭 jupyt	er Inv	estigatio	on_of_Seque	encing_Ru	n_Qualit	ty Last Checkp	point: a few se	conds ago (unsav	ed changes)			e
File Edit	View	Insert (Cell Kernel	Help								Python 3 O
8 + %	26	↑ ¥ H	C Cod	le 🗘	Cel	llToolbar						
	Inves	tigatio	on of Sec	quencir	ng Rui	n Quali	ty					
				differences be	tween the o	case and cont	trol conditions	s. However, these	e may have b	een influenced t	oy systemic	issues with
	our seque	ncer as sho	wit below.									
In [1]:	run_qual		= pandas.re	ad_csv("~/	Desktop/1	'runs_qualit	ty.csv")					
Out[1]:	Experi	mentNum	SampleName	RunDate	AverageQ	2						
	0 1		Expt1Sample1	2016-11-01	35							
	1 1		Expt1Sample2	2016-11-14	31	_						
	2 1		Expt1Sample3		32	_						
	3 2				27	_						
	4 2		Expt2Sample2		26	_						
	5 2		Expt2Sample3		22	_						
	6 3				20	_						
	7 3				21	_						
	8 3		Expt3Sample3	2017-01-31	20							
In [2]:	import n	atplotli	b.pyplot as		ues,run_c	guality_dat	ta["Average	eQ"],c=run_qu	uality_dat	a["Experimen	tNum"])	
Out[2]:	<matplot< td=""><td>lib.coll</td><td>ections.Path</td><td>Collection</td><td>at 0x10e</td><td>e277c88></td><td></td><td></td><td></td><td></td><td></td><td></td></matplot<>	lib.coll	ections.Path	Collection	at 0x10e	e277c88>						
	36		,	,	· _]							
	34 - 32 -				1							
	30	•••			1							
	28				-							
	26		0		-							
	24				1							
	22 - 20 -		•	. •	. 1							
	18	2		6								
				0	0							
In []:												





<pre> () @ Deatherst B992/networks/WrighProjects/Wrigh</pre>	Investiga	tior	n_of_Sequencing_R 3	<\								
Upty Cfr Investigation_of_Sequencing_Run_Quality Last Checkpoint: a few seconds app (insaved changes) P = Edit Verw Intert Call Kernel Help Plython 3 C P + R < P + R	C & 0 lo	cal	host:8892/notebo	oks/Work/Proje	cts/Investiga	tion_of_Sequ	encing_Rur	n_Quality.ipynl)		🚖 🖸 🤊 🍖 () 🔄 🛈 🗉 🚺
File Edit Verw Item Cell Kend Hep / Python 3 B Item Cell Item Cell Kend Hep / Python 3 B Item Cell Ite	ps 🙍 Google Cal	end	lar 🔥 Asana 🙆	Toggl report 🛛 🧧	a. AWS Sigr	n-In 🔋 AWS	Pricing	jupyter-genor	ics 😄 Terminal Co	ommands 🔇	Local Notebooks	» 🛅 Other Boo
<pre>> * * * * * * * * * * * * * * * * * *</pre>	🔁 jupyte	er	 Investigation 	on_of_Seque	encing_Ru	n_Quality	Last Check	point: a few se	conds ago (unsaved o	changes)		ۇ
<pre>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 or sequencer as show balow. fn [1; import pandar run,quality_data = pandaa.read_cev(*-/Desktop/runs_quality.cev*) run,quality_data Dest[1]: Tepriment/Max MampleName MuDate AverageD 1 ExptISample1 2016:11-01 35 1 ExptISample2 2016:11-41 31 2 ExptISample2 2016:11-41 31 2 ExptISample2 2016:11-41 31 2 ExptISample2 2016:11-42 30 2 ExptISample2 2016:11-01 42 3 ExptISample2 2017:01-02 28 3 ExptISample2 2017:01-02 28 5 ExptISa</pre>	File Edit	V	iew Insert	Cell Kernel	Help							Python 3 O
<pre>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 or sequencer as show balow. fn [1; import pandar run,quality_data = pandaa.read_cev(*-/Desktop/runs_quality.cev*) run,quality_data Dest[1]: Tepriment/Max MampleName MuDate AverageD 1 ExptISample1 2016:11-01 35 1 ExptISample2 2016:11-41 31 2 ExptISample2 2016:11-41 31 2 ExptISample2 2016:11-41 31 2 ExptISample2 2016:11-42 30 2 ExptISample2 2016:11-01 42 3 ExptISample2 2017:01-02 28 3 ExptISample2 2017:01-02 28 5 ExptISa</pre>	B + × 0	የነ	B + +)		le ¢	Cell	oolbar					
7 3 Expt3Sample2 2017-01-28 21 8 3 Expt3Sample3 2017-01-31 20 In [2]: import matplotlib inline import matplotlib.poplot as plt pl:.scatter(run_guality_data.index.values,run_guality_data["Average0"],c=run_guality_data["ExperimentNum"]) Out[2]: 6 9	In [1]: Out[1]:	Rec our imp run run 0 1 2 3 4 5	cent experiments : requencer as sho n_quality_data ExperimentNum 1 1 1 2 2 2	show significant i wm below: a = pandas.re SampleName Expt1Sample1 Expt1Sample2 Expt2Sample3 Expt2Sample2 Expt2Sample3	ad_csv("-/ RunDate 2016-11-01 2016-11-01 2016-12-03 2016-12-18 2017-01-02 2017-01-04	Desktop/r AverageQ 35 31 32 27 26 22	ase and con	atrol conditions	. However, these m	nay have be	en influenced by system	nic issues with
Out[2]: <pre> caseplotlib.collections.PathCollection at 0x10e277c88></pre>	In [2]:	8 %m	3 atplotlib inli port matplotli	Expt3Sample3	2017-01-31	20						
In []:		<ma 36 34 32 30 28 26 24 22 20</ma 	atplotlib.coll	ections.Path • • •				ta["Averag	90"],c=run_qual	lity_data	["ExperimentNum"]	3
	In []:											





🖲 🔍 🔵 📿 Investig	atio	n_of_Sequencing_R 3	× \						Θ
< → C ☆ 🛈 🗰	ocal	host:8892/notebo	oks/Work/Proje	cts/Investiga	tion_of_Seque	ncing_Run_Quality.ipynb		🖈 🗢 🤊 🚜 🔿	S 🕘 🕘 💭 📓 🗄
Apps 20 Google Ca	lend	lar 🔥 Asana 🎯	Toggl report 📑	a. AWS Sign	n-In 🔋 AWS F	ricing O jupyter-genomics	Terminal Commands		» 🗎 Other Bookmarks
📁 jupyt	eı	Investigatio	on_of_Seque	encing_Ru	n_Quality	ast Checkpoint: 7 minutes ag	o (unsaved changes)		e
File Edit	V	fiew Insert	Cell Kernel	Help					Python 3 O
8 + %	മ	B + +)	I C Cod	le \$	CellTo	lbar			
	Ir	vestigatio	on of See	quencii	ng Run	Quality			
				differences be	etween the cas	e and control conditions. He	owever, these may have	peen influenced by systemic	issues with
	ou	r sequencer as sho	wn below:						
In [1]:	ru	port pandas n_quality_dats n_quality_dats		ad_csv("~/	Desktop/ru	s_quality.csv")			
Out[1]:	Γ	ExperimentNum	SampleName	RunDate	AverageQ				
	0	1	Expt1Sample1	2016-11-01	35				
	1	1	Expt1Sample2	2016-11-14	31				
	2		Expt1Sample3	2016-12-03	32				
	3		Expt2Sample1	2016-12-18	27				
	4		Expt2Sample2		26				
	5		Expt2Sample3		22				
	6		Expt3Sample1	2017-01-23	20				
	7				21				
	8	3	Expt3Sample3	2017-01-31	20				
	im 	• • • • • • • • • • • • • • • • • • •	b.pyplot as		at 0x10e27	8088>			
TH []:									
_		_				_	_	_	_





🖲 🔵 🔵 📿 Investig	atio	n_of_Sequencing_R 3	< \						Θ
< → C ☆ 🗊 k	oca	host:8892/notebo	ooks/Work/Proje	cts/Investigat	ion_of_Sequencing_Run_	_Quality.ipynb		🖈 🗢 🕈 🍊 🔿	🧕 🛈 🧉 🗖 📕 E
					-In 🔋 AWS Pricing 🕥		C Terminal Commands		» 📄 Other Bookmarks
📁 jupyt	e	lnvestigatio	on_of_Seque	encing_Ru	n_Quality Last Checkp	point: a minute ago	(unsaved changes)		4
File Edit	\	/iew Insert	Cell Kernel	Help					Python 3 O
8 + %	മ	B + +)	H 🔳 C 😡	ie ¢	CellToolbar				
						. .			
		-		-	ng Run Qualit	-			
		cent experiments s r sequencer as sho		differences be	tween the case and cont	trol conditions. Ho	wever, these may have t	been influenced by systemic	issues with
In [1]:	in	port pandas							
		n_quality_data n_quality_data		ad_csv("~/	Desktop/runs_qualit	ty.csv")			
Out[1]:		ExperimentNum	SampleName	RunDate	AverageQ				
	0	1	Expt1Sample1	2016-11-01	35				
	1	1			31				
	2	1	Expt1Sample3		32				
	3		Expt2Sample1		27				
	4		Expt2Sample2		26				
	5				22				
	6		Expt3Sample1		20				
	7		Expt3Sample2		21				
	8	3	Expt3Sample3	2017-01-31	20				
In [2]:		atplotlib inli port matplotli		nlt					
					y_data["RunDate"]),	, run_quality_	data["AverageQ"],	":rs")	
Out[2]:	<11	atplotlib.coll	ections.Path	Collection	at 0x10e277c88>				
	3	•		,					
	3				1				
	3	•]				
	2								
	2	5 -	° •		-				
	2				-				
	2		۰						
	20			•	•				
		0 2	4	6	8				
In []:									





🗕 😐 🌔 📿 Investiga	itio	_of_Sequencing_R 3							θ
< → C ☆ 🛈 Ю	cal	host:8892/notebo	oks/Work/Proje	cts/Investiga	tion_of_Seq	uencing_Run_Quality.ipynl		* 🛡 🖣 🚜 🔿	🧕 🛈 🧉 🗖 📕 E
							ics 😄 Terminal Commands		» 🗎 Other Bookmarks
📁 jupyte	eı	Investigatio	on_of_Seque	encing_Ru	n_Quality	/ Last Checkpoint: a minute	ago (unsaved changes)		e
File Edit	٧	iew Insert	Cell Kernel	Help					Python 3 O
8 + %	ත	B + +)	l C Cod	ie 🗘	Cell	Toolbar			
	In	vestigation	on of See	quenciı	ng Rur	n Quality			
				differences be	tween the c	ase and control conditions	. However, these may have	been influenced by systemi	c issues with
	our	sequencer as sho	wn below:						
		port pandas							
		n_quality_data n_quality_data		ad_csv(~~/	Desktop/I	uns_quality.csv")			
Out[1]:	Π	ExperimentNum	SampleName	RunDate	AverageQ				
	0	1	Expt1Sample1	2016-11-01	35				
	<u> </u>	1	Expt1Sample2	2016-11-14	31				
		1	Expt1Sample3	2016-12-03	32				
	3			2016-12-18	27				
		2			26				
	5				22				
	6 7				20				
	7 8	3	Expt3Sample2 Expt3Sample3		21				
	•	3	Explosalliples	2017-01-31	20				
		atplotlib inli port matplotli		n]+					
					y_data["F	unDate"]), run_qual:	ty_data["AverageQ"],	":rs")	
Out[3]:	[<	natplotlib.lin	es.Line2D at	0x10e77ec	f8>]				
	36								
	34	N N			1				
	32	- and the second se	•		1				
	28]				
	26		•						
	24				-				
	22	-			_				
	20	Nov 08 20106v 22 20106	c 06 2006c 20 2036	in 03 201)8n 17 2	01 B n 31 2017				
In []:									
		_	_	_					





	Investiga	ition_of_Seque	encing_R ×						
<pre>c) upyyter Investigation_of_Sequencing_Run_Quality Last Obsolop(rit 3 minutes ago (inserted charges)</pre>	0 C C 0 10	calhost:889:	2/notebooks/Work/Proj	ects/Investigat	on_of_Sequenc	ng_Run_Quality.ipynb		🖈 🗢 🕈 🏀 🗘	S 0 👄 🗖 🛛
Image: Control of Sequencing Run Quality Image: Control of Sequencing Run Quality Control of Sequencing Run Quality.com Image: Control of Control of Sequencing Run Quality.com Image: Control of Cont	ps 🙍 Google Cal	endar 🛛 👶 As	ana 🔞 Toggl report 🚦	a, AWS Sign	In 🔋 AWS Pric	ing O jupyter-genomics	C Terminal Commands	C Local Notebooks	» 🗎 Other Book
<pre></pre>	📁 jupyte	er Inve	stigation_of_Sequ	encing_Ru	n_Quality Lat	t Checkpoint: 3 minutes ag	go (unsaved changes)		.
Image: contract of the contract	File Edit	View In	sert Cell Kernel	Help					Python 3 O
<pre>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: (n [1]:</pre>	B + ×	3 B 🛧		rkdown \$	CellToolt	ar			
<pre>run_quality_data = pandas.read_cev("-/Desktop/runs_quality.cev") run_quality_data = pandas.read_cev("-/Desktop/runs_quality.cev") Out[1]: ExperimentNum SampleName RunDate AverageO 0 1 Expt1Sample1 2016-11-01 35 1 1 Expt1Sample2 2016-11-41 31 2 1 Expt1Sample2 2016-12-03 32 3 2 Expt2Sample1 2016-12-18 27 4 2 Expt2Sample2 2017-01-02 26 5 2 Expt2Sample2 2017-01-02 20 7 3 Expt3Sample1 2017-01-28 20 8 3 Expt3Sample1 2017-01-28 20 7 3 Expt3Sample2 2017-01-28 20 8 3 Expt3Sample2 2017-01-28 20 8 3 Expt3Sample2 2017-01-28 20 7 3 Expt3Sample2 2017-01-28 20 8 3 Expt3Sample2 2017-01-28 20 7 4 20 4 20 4 20 4 20 4 20 4 4 4 4 4 4 4</pre>		Recent expe	riments show significant	-	-	-	owever, these may have	been influenced by systemic	; issues with
<pre> Low number of any set of a se</pre>		run_qualit	ty_data = pandas.r	ead_csv("~/)	esktop/runs	guality.csv")			
<pre>i 1</pre>	Out[1]:	Experime	entNum SampleName	RunDate	AverageQ				
<pre> 2 1 Expt1Sample 2016-12-03 32 3 2 Expt2Sample 2016-12-18 27 4 2 Expt2Sample 2017-01-02 26 8 2 Expt2Sample 2017-01-02 20 7 3 Expt3Sample 2017-01-28 21 8 3 Expt3Sample 2017-01-28 21 8 3 Expt3Sample 2017-01-31 20 In [3]: mapter matplotlib inline mpoor matplotlib.plot(panda.s.c_datetime(run_quality_data[*RunDate*]), run_quality_data[*XverageO*], *:rs*) Out[3]: [<matplotlib.lines.line2d 0x10077ecf8="" at="">] 0ut[3]: [<matplotlib.lines.line2d 0x10077ecf8="" at="">] </matplotlib.lines.line2d></matplotlib.lines.line2d></matplotlib.lines.line2d></matplotlib.lines.line2d></matplotlib.lines.line2d></matplotlib.lines.line2d></matplotlib.lines.line2d></pre>			Expt1Sample1	2016-11-01	35				
<pre> a 2 Expt2Sample2 2016-12-18 27 b 2 Expt2Sample2 2017-01-02 26 b 2 Expt2Sample2 2017-01-02 22 b 3 Expt3Sample2 2017-01-04 22 b 3 Expt3Sample2 2017-01-28 21 b 3 Expt3Sample2 2017-01-28 21 b 3 Expt3Sample2 2017-01-28 21 c 1 3]; imstplotlib inline import mstplotlib.plot g plt plt.plot(pandas.to_datetime(run_guality_date["hunDate"]), run_guality_data["Average0"], ":rs") Out[3]; (=matplotlib.inlee.Line2D at 0x1007recf8>) a b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 22 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 200Er: 06 200Hr 20 2014 n 1720[An 31 2017 b 2 Outer v2 0 Outer v2</pre>			Expt1Sample2	2016-11-14	31				
<pre></pre>			Expt1Sample3	2016-12-03	32				
<pre></pre>			Expt2Sample1	2016-12-18	27				
6 3 Expt3Sample1 2017-01-23 20 7 3 Expt3Sample2 2017-01-28 21 8 3 Expt3Sample3 2017-01-31 20 In [3]: tmatplotlib inline import matplotlib.ipplot as plt pic.plot(pandes.co_datetime(run_quality_data[*RunDate"]), run_quality_data[*Average("], ":rs") Out[3]: [<matplotlib.ines.line2d 0x10e77ecf8="" at="">] 3 5 5 4 5 5 5 5 5 5 6 5 7 5 <td></td><td></td><td>Expt2Sample2</td><td>2017-01-02</td><td>26</td><td></td><td></td><td></td><td></td></matplotlib.ines.line2d>			Expt2Sample2	2017-01-02	26				
i 3 Expt3Sample2 2017-01-28 21 i 3 Expt3Sample3 2017-01-28 21 in (3): Immort matplotlib inline immort matplotlib.pot(pandas.co.datetime(run.quality_data[*Aubate*]), run_quality_data[*Average0*], *:rs*) Out[3]: (<matplotlib.lines.line2d 0x10077ecf8="" at="">] 0 (<matplotlib.lines.line2d 0x10077ecf8="" at="">] 0 0</matplotlib.lines.line2d></matplotlib.lines.line2d>			Expt2Sample3	2017-01-04	22				
a Expt3Sample3 2017-01-31 In [3]: Imaption interpretation of the set of the s			Expt3Sample1	2017-01-23	20				
<pre>In [3]: imatplotlib inline import mstplotlib.pyplot as plt plt.plot(pandas.to_datatime(run_quality_data["RunDate"]), run_quality_data["Average("], ":rs") Out[3]: [<mstplotlib.lines.line2d 0x10e77ecf8="" at="">]</mstplotlib.lines.line2d></pre>					21				
<pre>import matplotlib.pplot as plt plot(pandas.co_datetime(run_quality_data["Auverage0"], ":rs") Out[3]; [<matplotlib.lines.line2d 0x10077ecf8="" at="">] 0ut[3]; [<matplotlib.lines.line2d 0x10077ecf8="" at="">] 0ut[3]; [<matplotlib.government of="" state="" td="" th<="" the=""><td></td><td>8 3</td><td>Expt3Sample3</td><td>2017-01-31</td><td>20</td><td></td><td></td><td></td><td></td></matplotlib.government></matplotlib.lines.line2d></matplotlib.lines.line2d></pre>		8 3	Expt3Sample3	2017-01-31	20				
36 37 39 39 30 39 30 30 30 30 30 30 30 30 30 30		<pre>import mat plt.plot(p)</pre>	tplotlib.pyplot as pandas.to_datetime	(run_quality		ute"]), run_quality	_data["AverageQ"],	":rs")	
20 26 24 22 20 Nov 68 2006: 06 2066: 20 2016: 01 2010: 01 2010: 01 2010: 01 2017	out[3]:	36 34 32	••••••••••••••••••••••••••••••••••••••		-				
24 22 20 Nov 08 2016: 05 2016: 05 2016: 05 2016: 03 2013: 17 2013: 31 2017		28			-				
22 20 Nov 08 2016+ 22 2016+ 06 2016+ 03 2016h 13 2016h 13 2018h 31 2017		26			-				
²⁰ Nov 08 2006v 22 2016c 05 2006c 20 20160 03 20180 37 20100 31 2017		24 -							
		22 -		- Andrewski (* 1997)					
Conclusion: Differences between experiments are likely conflated with run quality issues.		20	6v 22 20 Dac 06 20 Dac 20 20]	á n 03 201 j án 17 20	1 Ji n 31 2017				
	[**Conclus:	ion**: Differences	between exp	eriments ar	e likely conflated	with run quality i	ssues.	





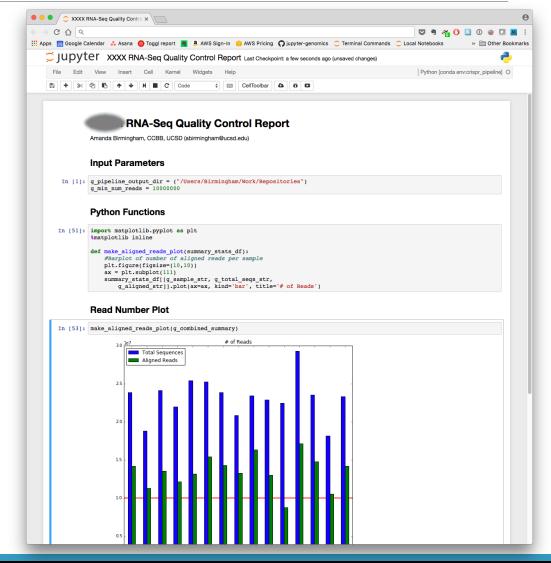
<pre> Output: Set in the set</pre>	🖲 😑 🤇 🔿 Investiga	atio	n_of_Sequencing_R 3	×						Θ
<pre>[exect @ code deciders</pre>	< → C ☆ © k	cal	host:8892/notebo	ooks/Work/Proje	cts/Investiga	tion_of_Sequencing	g_Run_Quality.ipynb		🖈 🗢 🖣 🍊 🕐	S () 👄 🗖 M :
<pre>Prince Description: Prince Description: P</pre>								Terminal Commands		
<code-block></code-block>	📁 jupyt	eı	Investigatio	on_of_Seque	encing_Ru	In_Quality Last (Checkpoint: 3 minutes ag	o (unsaved changes)		4
<section-header> Control of the control of the</section-header>	File Edit	V	iew Insert	Cell Kernel	Help					Python 3 O
<pre>Best experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with susception to be a strone below: (i (1):</pre>	8 + %	ሪ	6 + +)	H 🔳 C' Coc	ie ¢	CellToolbar				
<pre>Best experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with susception below: (i (1):</pre>										
<pre>Best experiments show significant differences between the case and control conditions. However, these may have been influenced by systemic issues with susception to be a strone below: (i (1):</pre>										
<pre>or sequencer as shown below: f 1 {1; } f = part pandes</pre>		In	vestigatio	on of Se	quencii	ng Run Qu	ality			
<pre>or sequencer as shown below: f 1 {1; } f = part pandes</pre>			-		-	-	-	owever, these may have b	een influenced by systemic	issues with
<pre>run_quality_data = pandas.read_cev(*-/Desktop/runs_quality.cev*) run_quality_data Out[1]:</pre>										
<pre>run_quality_data = pandas.read_cev(*-/Desktop/runs_quality.cev*) run_quality_data Out[1]:</pre>	In [1]:	im	port pandas							
<pre>superintentive appearate intentive appearate intentintentive appearate intentive appearate intentive</pre>		ru	n_quality_data		ad_csv("~/	Desktop/runs_g	uality.csv")			
<pre>i i i Expt1Sample1 2016-11-14 ji i Expt1Sample2 2016-12-03 ji j 2 Expt2Sample1 2017-01-02 ji j 2 Expt2Sample2 2017-01-02 ji j 2 Expt2Sample2 2017-01-02 ji j 3 Expt3Sample2 2017-01-03 ji j 3 Expt3Sample2 2017-01-23 ji j 4 j 2 Expt3Sample2 2018-01 ji j 4</pre>	Out[1]:	Π	ExperimentNum	SampleName	RunDate	AverageQ				
<pre></pre>		0	1	Expt1Sample1	2016-11-01	35				
<pre>i</pre>		1	1	Expt1Sample2	2016-11-14	31				
<pre></pre>		2	1	Expt1Sample3	2016-12-03	32				
i i			2	Expt2Sample1	2016-12-18	27				
a Expt3Samplei 2017-01-28 20 a Expt3Samplei 2017-01-28 21 b a Expt3Samplei 2017-01-28 21 b a Expt3Samplei 2017-01-28 21 b a Expt3Samplei 2017-01-28 21 c a Expt3Samplei 2017-01-28 21 b a Expt3Samplei 2017-01-28 21 c a Expt3Samplei 2017-01-28 21 f f a Expt3Samplei 2017-01-28 21 f f f f f f f f f f a Expt3Samplei f				Expt2Sample2	2017-01-02	26				
<pre>import matplotlib inline import matplotlib.pyplot as plt plt.plot(plusmas.to_datatione(run_quality_data["AunDate"]), run_quality_data["AverageQ"], ";rs") Out[3]: ["matplotlib.lines.Line2D at 0x10e77ecf8>] 0ut[3]: ["matplotlib.lines.Line2D at 0x10e77ecf8>]</pre>				Expt2Sample3	2017-01-04	22				
<pre>a</pre>		\vdash	3	Expt3Sample1	2017-01-23	20				
<pre>In [3]: imaplotlib inlime import matplotlib.pyplot as plt plt.plot(pandas.to_datetime(run_guality_data(*AunDate*)), run_guality_data(*AverageQ*), *:rs*) Out[3]: (satplotlib.lines.Line2D at 0x10e77ecf8>)</pre>				Expt3Sample2						
<pre>import matplotlib.pyplet as plt plt.plot(pandas.to_datesime(rum quality_data["RunDate"]), run_quality_data["AverageQ"], ":rs") Out[]: ["matplotlib.lines.Line2D at 0x10e77ecf8>]</pre>		8	3	Expt3Sample3	2017-01-31	20				
<pre>import matplotlib.pyplet as plt plt.plot(pandas.to_datesime(rum quality_data["RunDate"]), run_quality_data["AverageQ"], ":rs") Out[]: ["matplotlib.lines.Line2D at 0x10e77ecf8>]</pre>	In [3]:	8m	atplotlib inli	ne						
Out[3]: [<pre>(matplotlib.lines.Line2D at 0x10e77ecf8>]</pre>						v data["RunDat	e"l), run quality	data["Average0"].	":rs")	
The second secon				-			e j), iun_quuitej	_uucu[morageg];		
Conclusion: Differences between experiments are likely conflated with run quality issues.	000(0)1									
no n		34	N							
Rev 08 2006v 22 201Ec 06 200Ec 20 201an 03 201an 31 2017 Conclusion: Differences between experiments are likely conflated with run quality issues.		32	\sim							
Conclusion: Differences between experiments are likely conflated with run quality issues.		30	-			-				
24 22 ³⁰ Nov di 2016c di 2016c di 2016c di 2016c 11 2016 ni 1 2016 ni 1 2016 ni 1 2017 Conclusion: Differences between experiments are likely conflated with run quality issues.		28	-	$\sim \sim$		-				
22 ²⁰ Nov dž 2086v 22 2018c dž 2086c 20 2018n 01 2018n 11 2018n 13 2017 Conclusion: Differences between experiments are likely conflated with run quality issues.		26	i -	- Contract		-				
²⁰ Nov dž 2016v 22 2016c dž 2016e 20 2016n 03 2016n 13 2016n 13 2017 Conclusion: Differences between experiments are likely conflated with run quality issues.		24	-			-				
Conclusion: Differences between experiments are likely conflated with run quality issues.		22	-		- Anna Anna Anna Anna Anna Anna Anna Ann					
		20	Nov 08 20106v 22 20106	tc 06 2006c 20 2036	n 03 201 j ên 17 2	01fin 31 2017				
		Co	nclusion: Differen	ces between exp	periments are	likely conflated with	n run quality issues.			
An Up	In []:									
										-





Jupyter Notebooks: Friend or Foe?

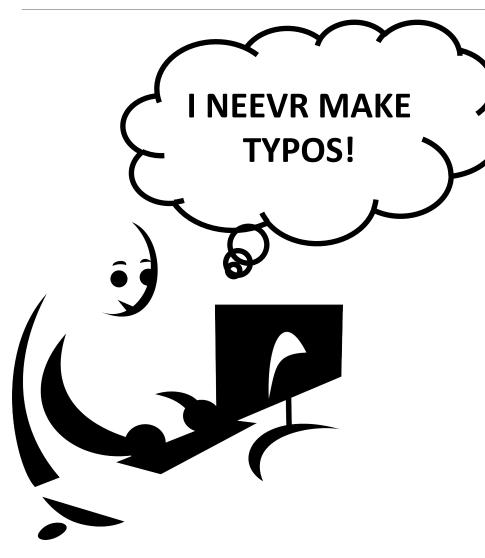
- Are notebooks the key to reproducibility?
 - Data Carpentry offers an entire workshop on "Reproducible Research using Jupyter Notebooks"
- Easy to save, modify, and extend
 - Great for rerunning or tweaking previous data analyses
- CCBB delivers analyses as notebooks
 - Report becomes more than a record—it 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







(Inter-)Actively Dangerous

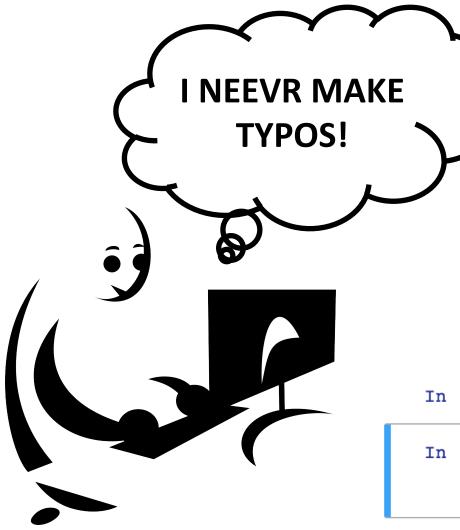


- Interactivity can also be a *huge danger* to reproducibility
- Humans are inconsistent
 - We make unpredictable mistakes
 - Thus, "interactive"="bad" for repetitive tasks
 - Like primary NGS analysis pipelines
- Jupyter Notebooks can be inconsistent, too
 - Changing code/variables in a notebook does 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 \odot
- 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

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 \odot
- 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

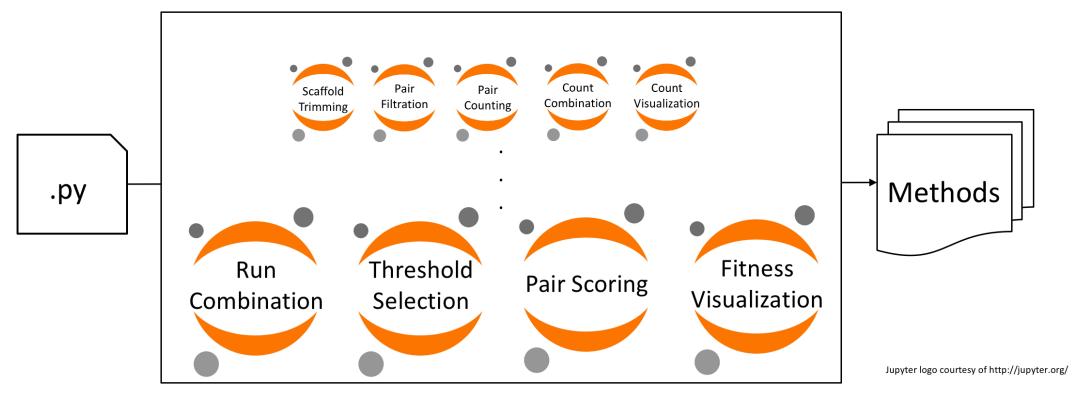
- A sample NGS pipeline using Jupyter Notebooks
 - Goal: identify gene pairs with synergistic survival effects (positive or negative)
 - Experimental system:: Dual-gene knock-outs in human cell lines using CRISPR
 - Read-out: number of instances of each CRISPR guide in final population, assessed by NGS





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







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



<u>UC San Diego</u>

Altman Clinical and Translational Research Institute



