Reproducible computing for your own benefit

Ming (Tommy) Tang

Senior Scientist at Dana-Farber Cancer Institute

Twitter: tangming2005

Blog: https://divingintogeneticsandgenomics.rbind.io/

Reproducibility crisis

Every baby knows the scientific method!
1 Make an observation.
2 Form a hypothesis.
3 Perform the experiment.
4 Analyze the data.
5 Report your findings.
6 Invite others to reproduce the results.

http://biobungalow.weebly.com/bio-bungalow-blog/everybody-knows-the-scientific-method

Most computational research is not reproducible.

I don’t know of a systematic study, but of papers that I read, approximately 95% fail to include details necessary for replication.

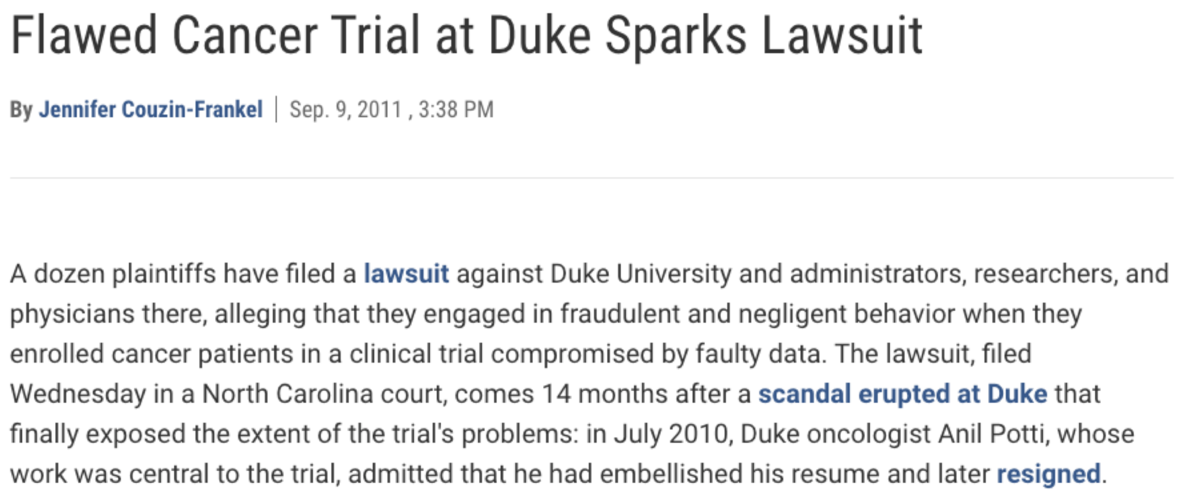
It’s very hard to build off of research like this.

(There’s a lot more to say about repeatability, reproducibility and replicability than I can fit in here…)

Titus Brown

An example

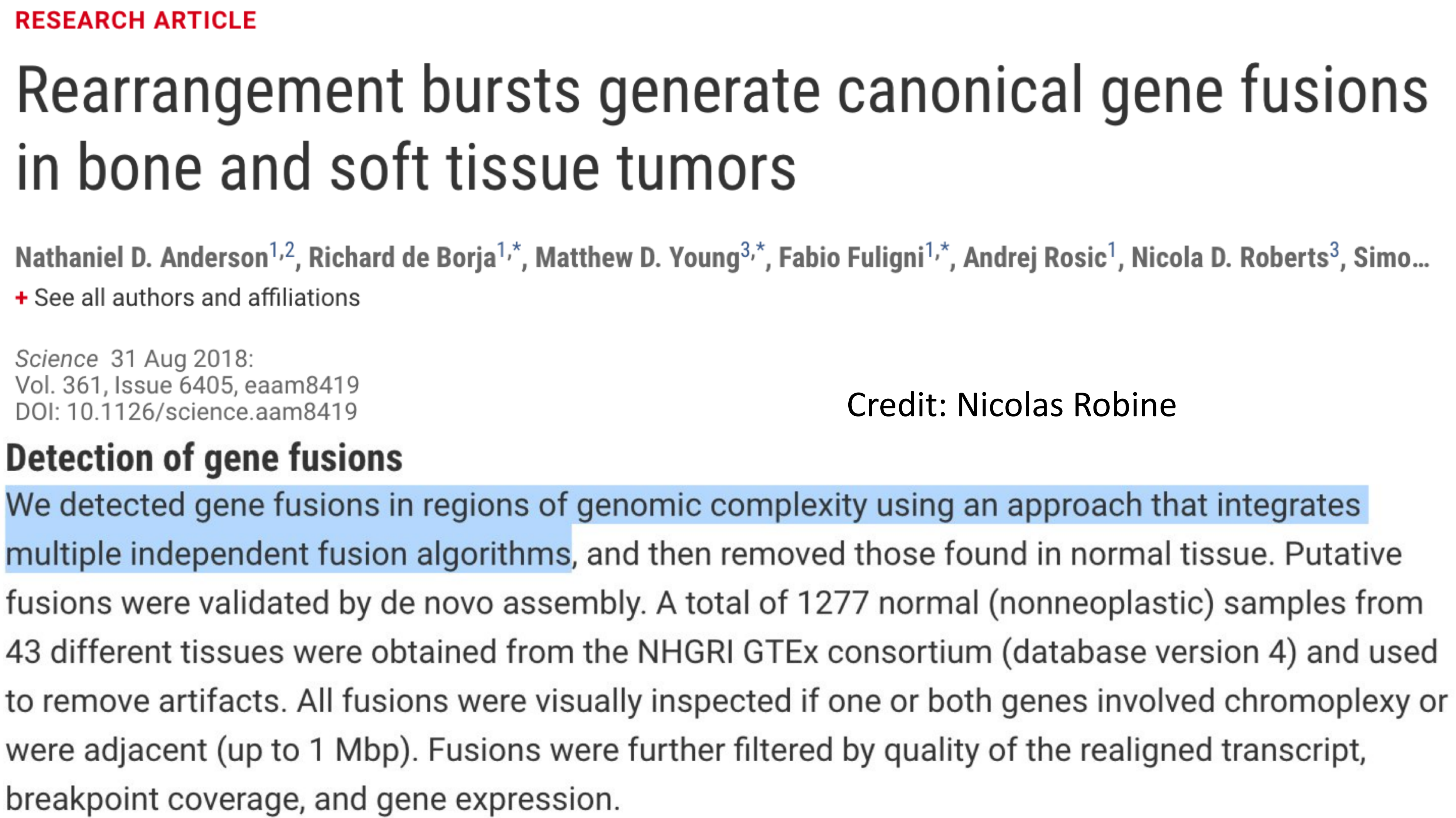
- The Importance of Reproducible Research in High-Throughput Biology.
- https://www.youtube.com/watch?v=7gYIs7uYbMo
- By Dr.Keith A. Baggerly from MD Anderson Cancer center.
- Highly recommend, Keith is very fun.


Flawed Cancer Trial at Duke Sparks Lawsuit

By Jennifer Couzin-Frankel | Sep. 9, 2011 , 3:38 PM

A dozen plaintiffs have filed a lawsuit against Duke University and administrators, researchers, and physicians there, alleging that they engaged in fraudulent and negligent behavior when they enrolled cancer patients in a clinical trial compromised by faulty data. The lawsuit, filed Wednesday in a North Carolina court, comes 14 months after a scandal erupted at Duke that finally exposed the extent of the trial's problems: in July 2010, Duke oncologist Anil Potti, whose work was central to the trial, admitted that he had embellished his resume and later resigned.

Method matters



RESEARCH ARTICLE

Rearrangement bursts generate canonical gene fusions in bone and soft tissue tumors

Nathaniel D. Anderson[1,2], Richard de Borja[1,*], Matthew D. Young[3,*], Fabio Fuligni[1,*], Andrej Rosic[1], Nicola D. Roberts[3], Simo…

+ See all authors and affiliations

Science 31 Aug 2018:
Vol. 361, Issue 6405, eaam8419
DOI: 10.1126/science.aam8419

Credit: Nicolas Robine

Detection of gene fusions

We detected gene fusions in regions of genomic complexity using an approach that integrates multiple independent fusion algorithms, and then removed those found in normal tissue. Putative fusions were validated by de novo assembly. A total of 1277 normal (nonneoplastic) samples from 43 different tissues were obtained from the NHGRI GTEx consortium (database version 4) and used to remove artifacts. All fusions were visually inspected if one or both genes involved chromoplexy or were adjacent (up to 1 Mbp). Fusions were further filtered by quality of the realigned transcript, breakpoint coverage, and gene expression.

Why reproducibility is hard?

Why reproducibility is hard?

- 1. lack of method description.
- 2. versions of the tools are different. (e.g. R/python/bioinformatics tools)
- 3. different machines (unix vs windows).

If it is so hard, should you care?

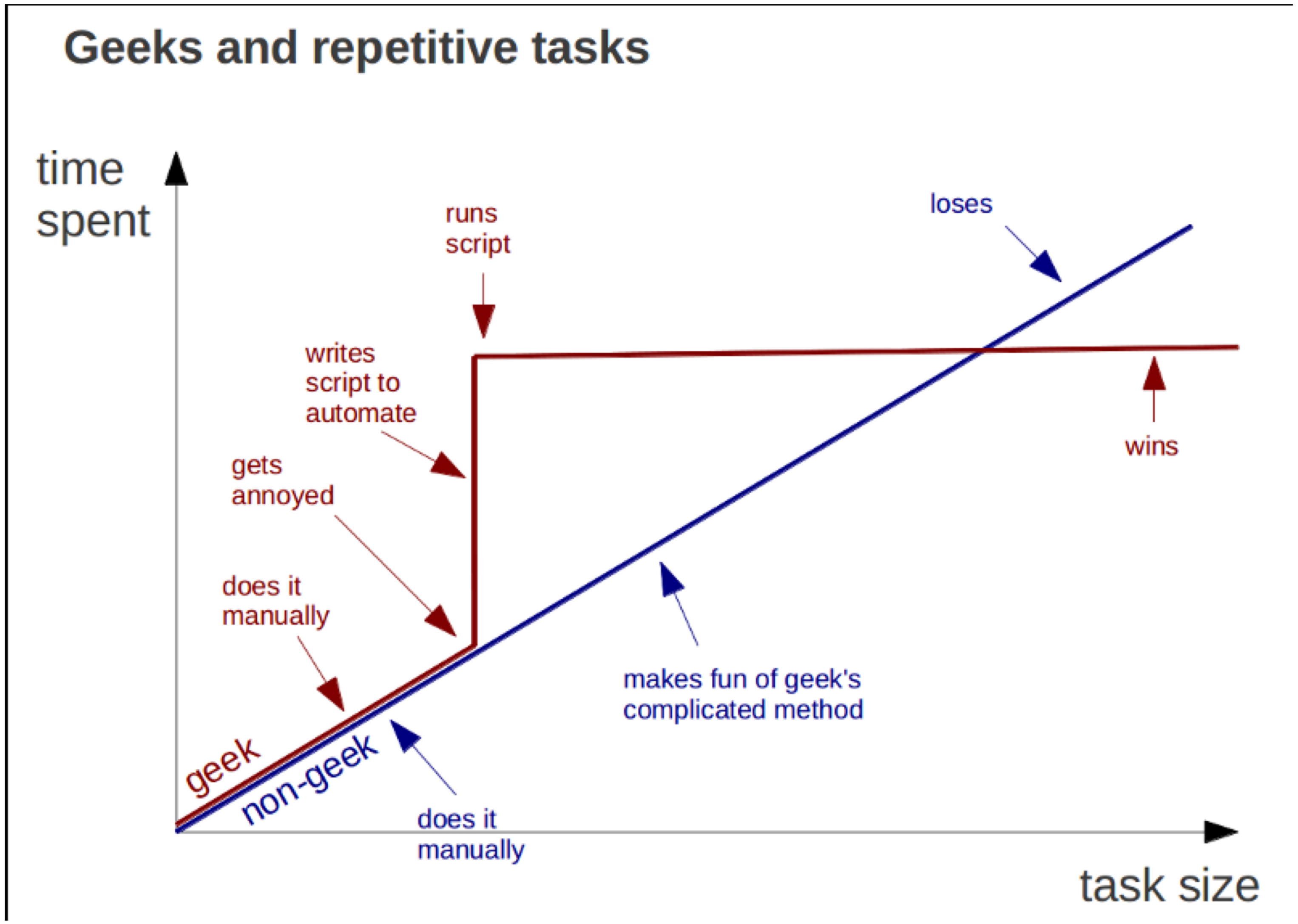
- Keep this in mind: You are going to do the same analysis for sure in the future yourself!
- This is for your own benefit.

How to ensure reproducibility

- Automation (write script to do the tasks, avoid manual editing)
- Git version control code and data.
- Jupyter/R Notebook, documentation
- Containers (docker, singularity, biocontainers
- https://biocontainers.pro/)

- https://mangul-lab-usc.github.io/enhancing_reproducibility/
- https://academic.oup.com/gigascience/article/9/6/giaa056/5849489

Automation saves you time in the long run



Computers are good at repetitive work

Side effect of automation

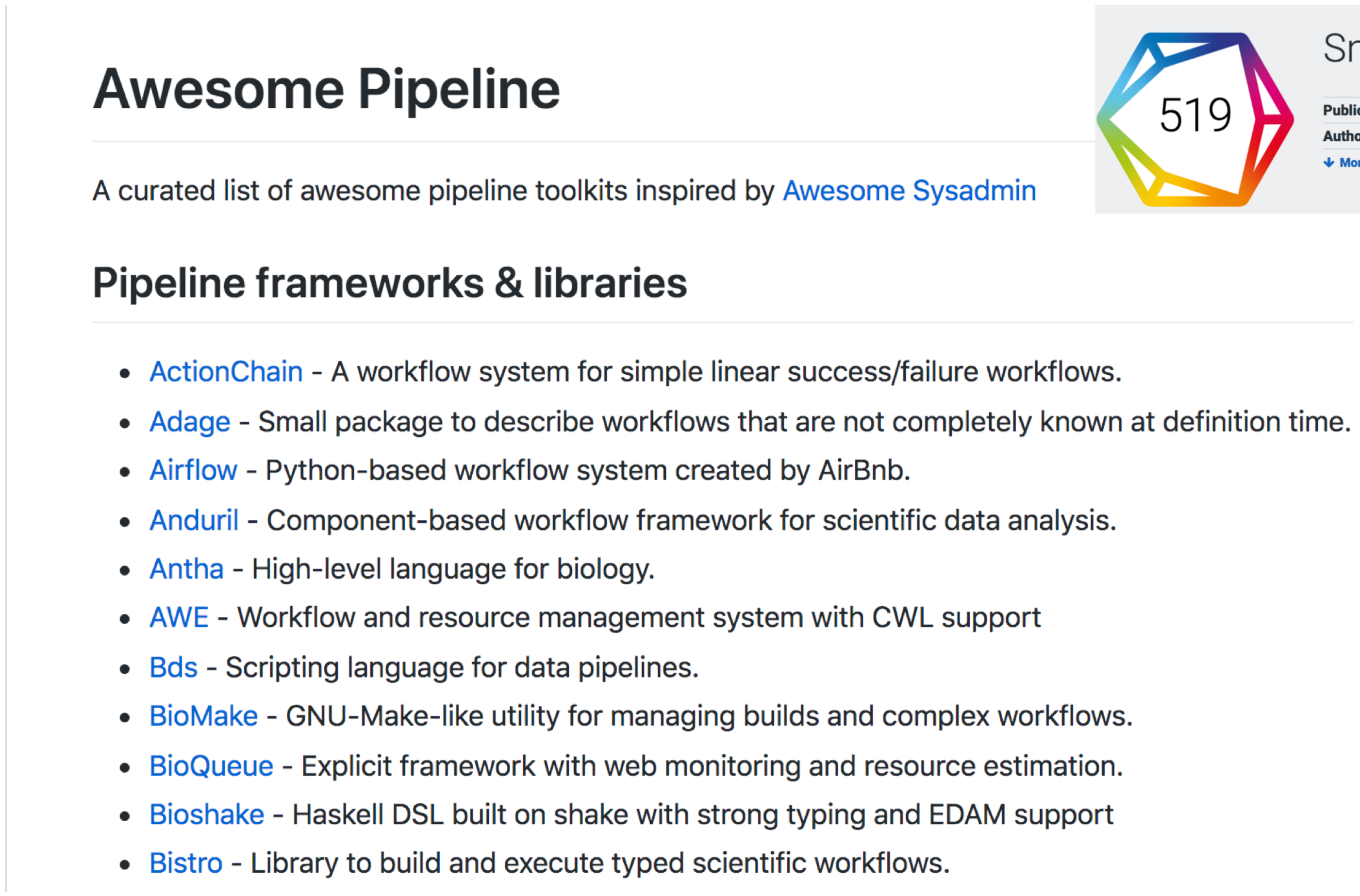
- The best documentation is automation
- Write scripts for everything unless it is not possible. (manual editing, document!)

Credit to someone in the twitter-verse ☺

Tips for automation

- 1. if you have a repetitive simple task, put them in to a shell script: my_routine.sh.
- 2. good old make
- 3. more recent snakemake, nextflow, WDL etc.

Many workflow languages/engines

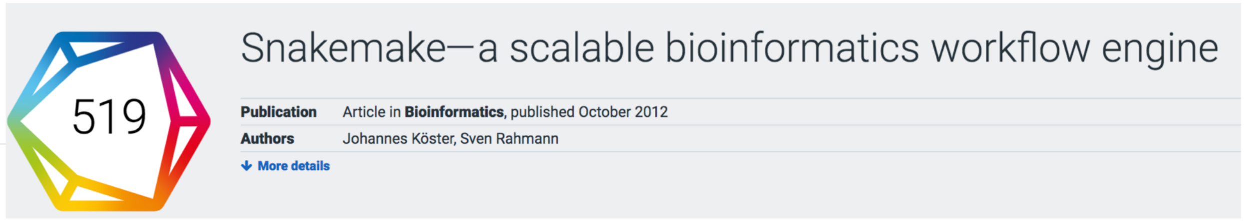


Awesome Pipeline

A curated list of awesome pipeline toolkits inspired by Awesome Sysadmin

Pipeline frameworks & libraries

- ActionChain - A workflow system for simple linear success/failure workflows.
- Adage - Small package to describe workflows that are not completely known at definition time.
- Airflow - Python-based workflow system created by AirBnb.
- Anduril - Component-based workflow framework for scientific data analysis.
- Antha - High-level language for biology.
- AWE - Workflow and resource management system with CWL support
- Bds - Scripting language for data pipelines.
- BioMake - GNU-Make-like utility for managing builds and complex workflows.
- BioQueue - Explicit framework with web monitoring and resource estimation.
- Bioshake - Haskell DSL built on shake with strong typing and EDAM support
- Bistro - Library to build and execute typed scientific workflows.



519

Snakemake—a scalable bioinformatics workflow engine

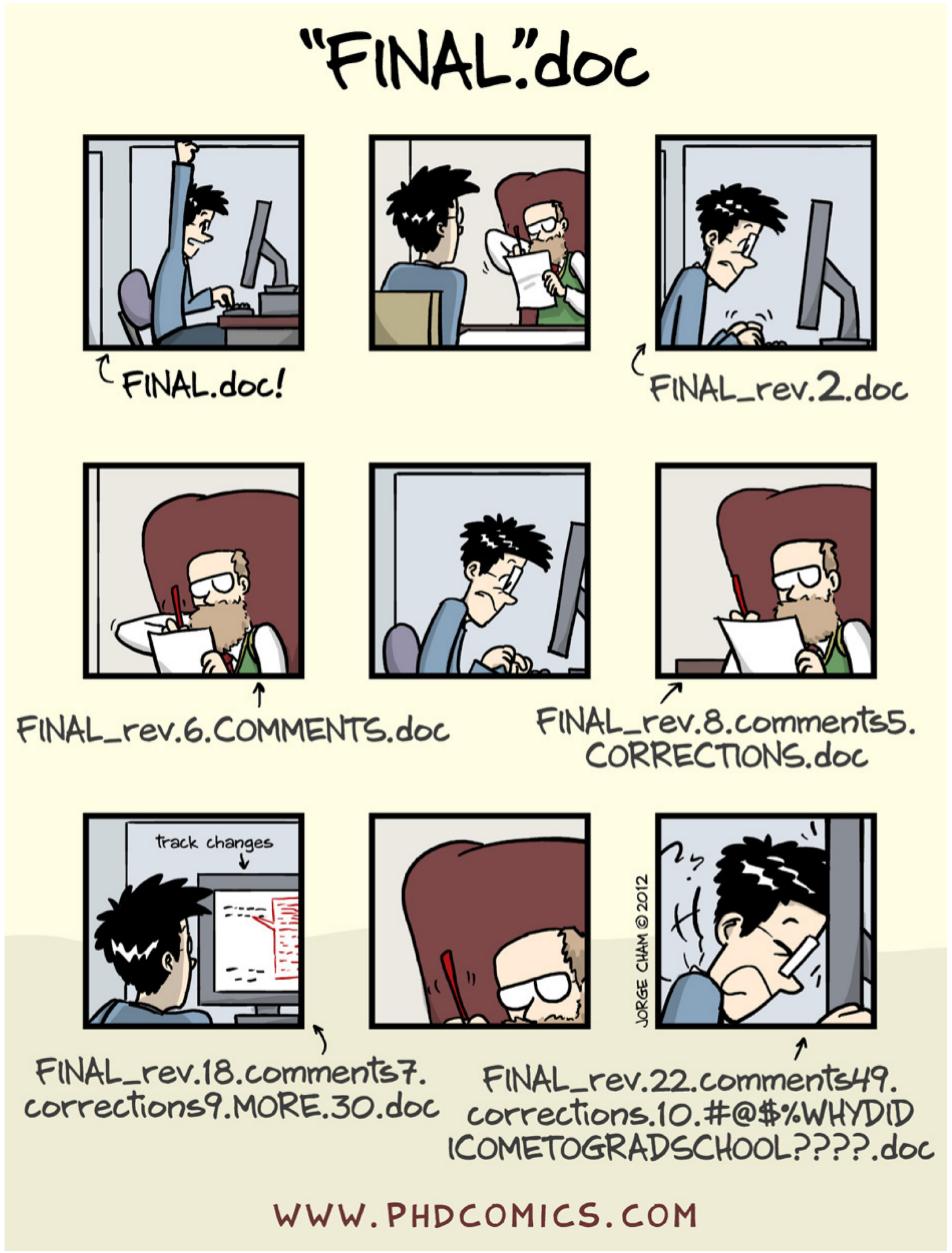
Publication	Article in **Bioinformatics**, published October 2012
Authors	Johannes Köster, Sven Rahmann

More details

{wdl}

nextflow

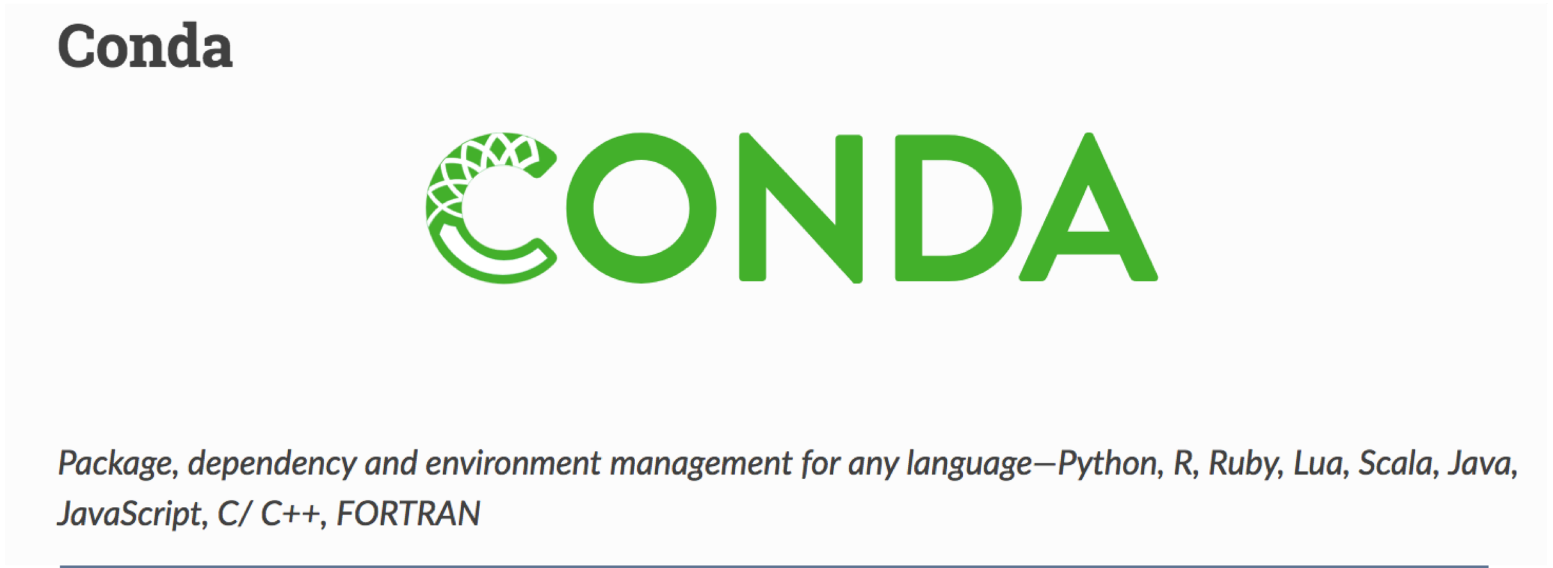
https://github.com/pditommaso/awesome-pipeline


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDIDICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

Version control

- Git
- Github
- Gitlab

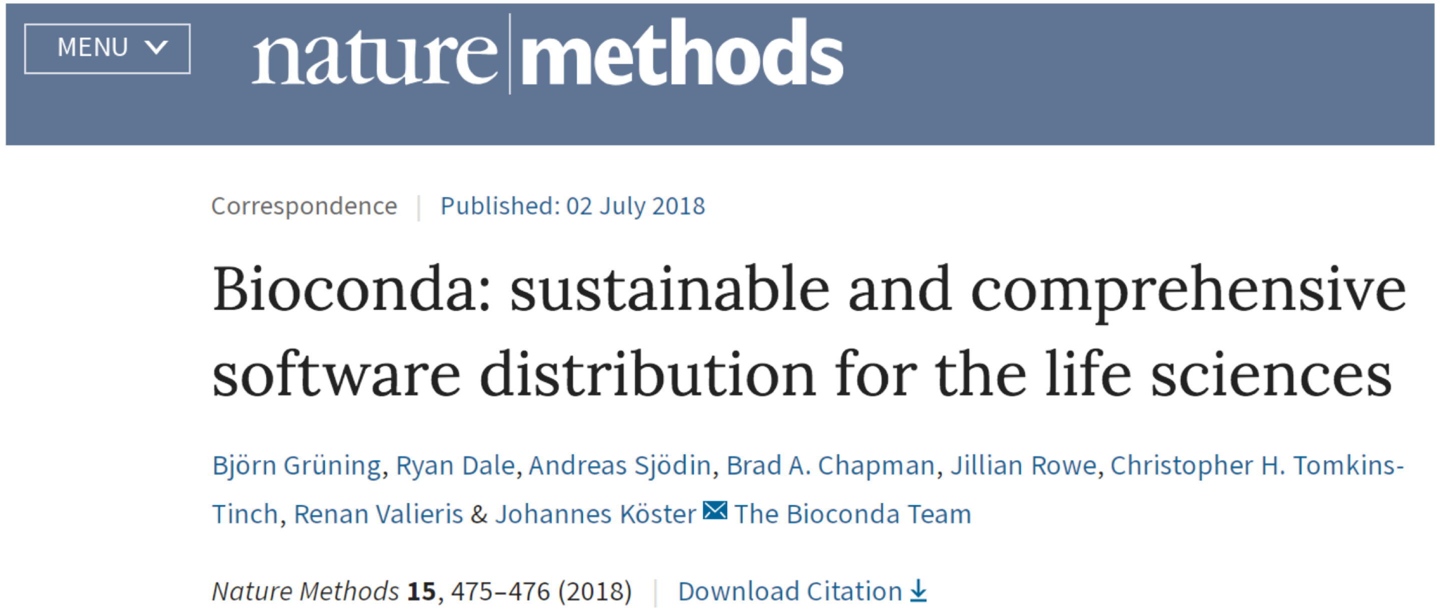
conda and biocoda



Conda

CONDA

Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN



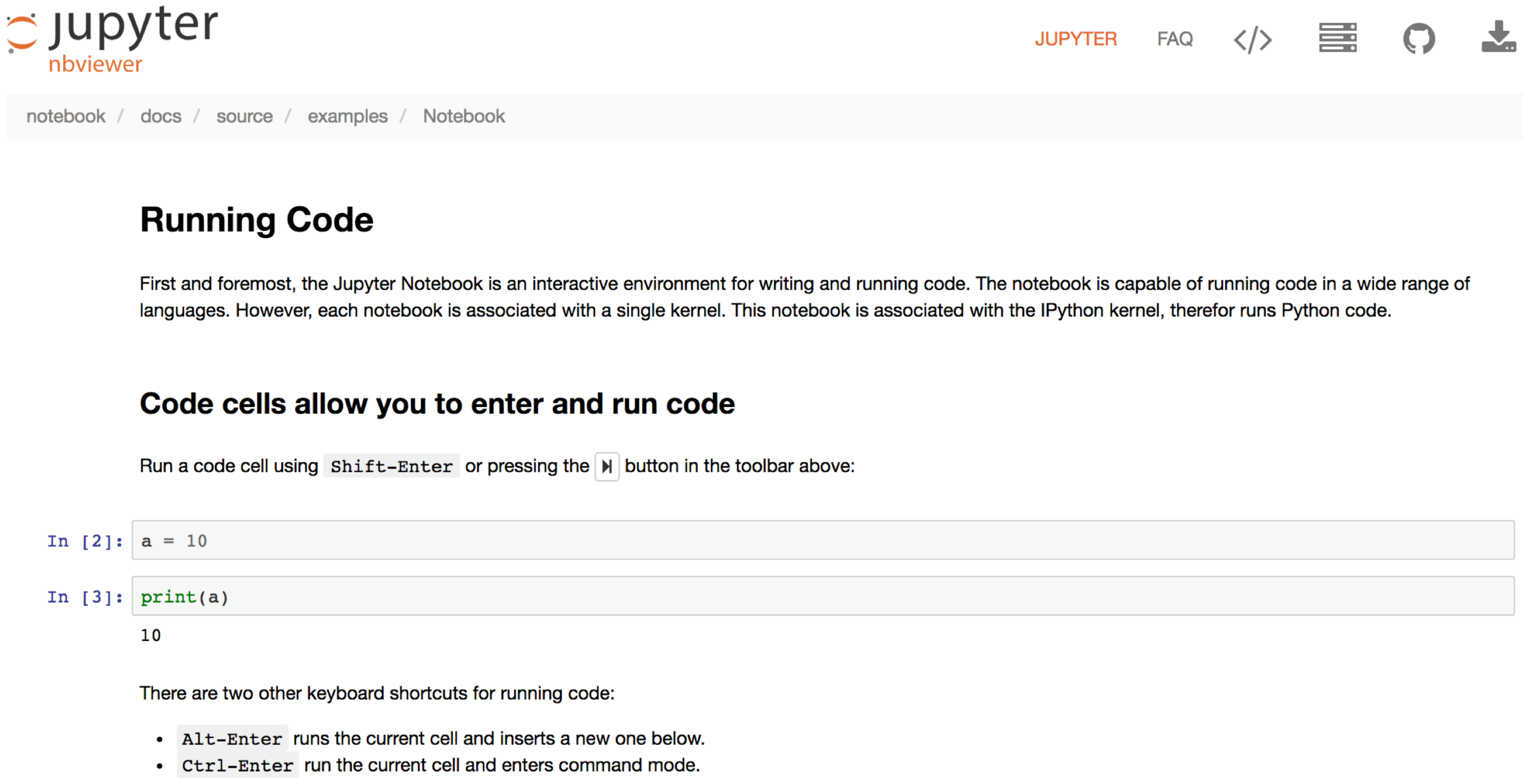
Correspondence | Published: 02 July 2018

Bioconda: sustainable and comprehensive software distribution for the life sciences

Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A. Chapman, Jillian Rowe, Christopher H. Tomkins-Tinch, Renan Valieris & Johannes Köster The Bioconda Team

Nature Methods **15**, 475–476 (2018) | Download Citation

Jupyter Notebooks



Running Code

First and foremost, the Jupyter Notebook is an interactive environment for writing and running code. The notebook is capable of running code in a wide range of languages. However, each notebook is associated with a single kernel. This notebook is associated with the IPython kernel, therefor runs Python code.

Code cells allow you to enter and run code

Run a code cell using `Shift-Enter` or pressing the ⏯ button in the toolbar above:

```
In [2]: a = 10
```

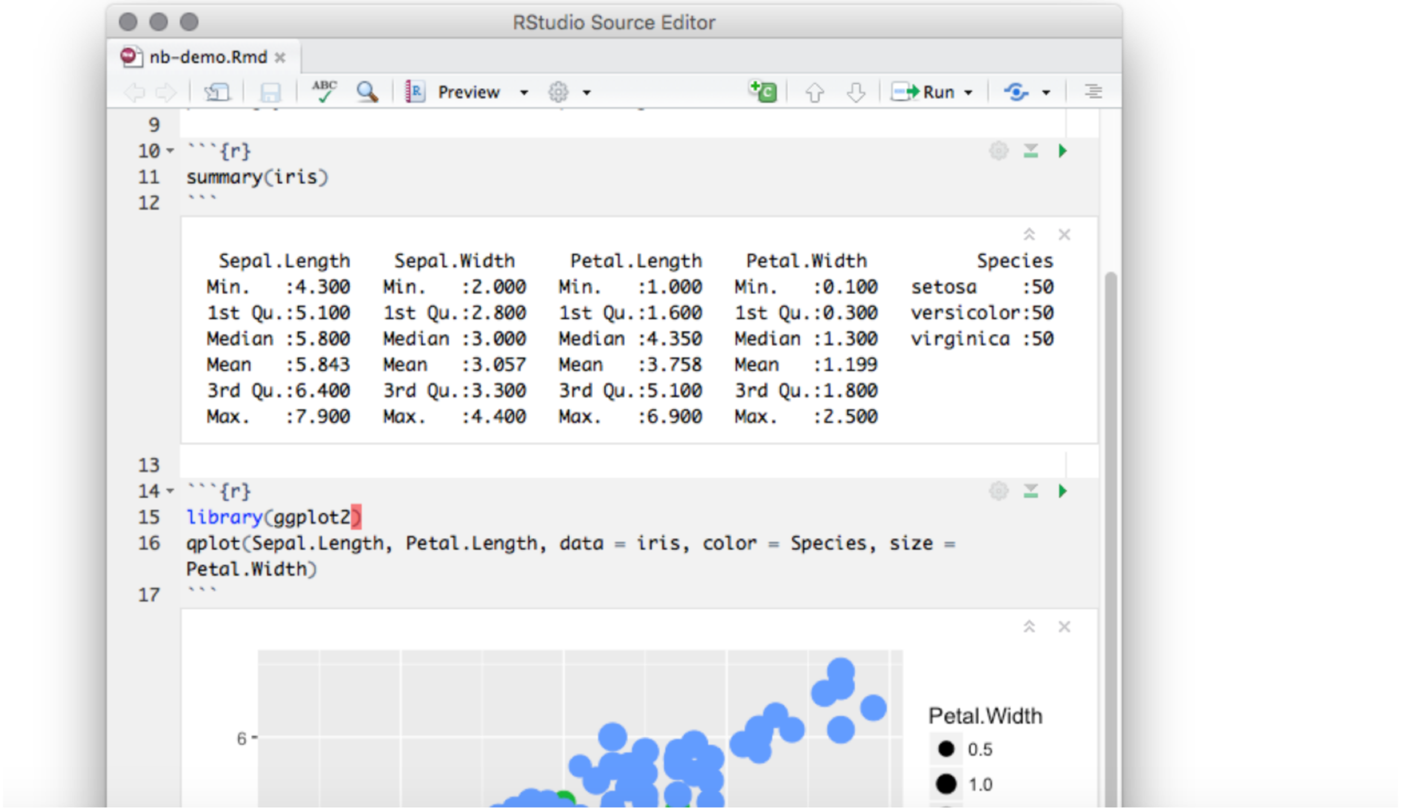
```
In [3]: print(a)

10
```

There are two other keyboard shortcuts for running code:

- `Alt-Enter` runs the current cell and inserts a new one below.
- `Ctrl-Enter` run the current cell and enters command mode.

R notebook/Rmarkdown

An R Notebook is an R Markdown document with chunks that can be executed independently and interactively, with output visible immediately beneath the input.



~~~
```{r}
summary(iris)
```
~~~

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

~~~
```{r}
library(ggplot2)
qplot(Sepal.Length, Petal.Length, data = iris, color = Species, size =
Petal.Width)
```
~~~

Docker/singularity

- Why docker?
- Imagine you are working on an analysis in R and you send your code to a friend. Your friend runs exactly this code on exactly the same data set but gets a slightly different result. This can have various reasons such as a different operating system, a different version of an R package, etc. Docker is trying to solve problems like that.
- This just happened between me and Brandon!

https://cyverse-cybercarpentry-container-workshop-2018.readthedocs-hosted.com/en/latest/docker/dockerintro.html
https://ropenscilabs.github.io/r-docker-tutorial/01-what-and-why.html

Blog posts on how to use them:
https://divingintogeneticsandgenomics.rbind.io/post/run-rstudio-server-with-singularity-on-hpc/
https://divingintogeneticsandgenomics.rbind.io/post/develop-bioconductor-packages-with-docker-container/

Other important untaught skills

- Naming files
- Use excel wisely
- Data/Project organization
- backup plans

Naming files is not easy

NO

myabstract.docx
Joe’s Filenames Use Spaces and Punctuation.xlsx
figure 1.png
fig 2.png
JW7d^(2sl@deletethisandyourcareerisoverWx2*.txt

YES

2014-06-08_abstract-for-sla.docx
joes-filenames-are-getting-better.xlsx
fig01_scatterplot-talk-length-vs-interest.png
fig02_histogram-talk-attendance.png
1986-01-28_raw-data-from-challenger-o-rings.txt

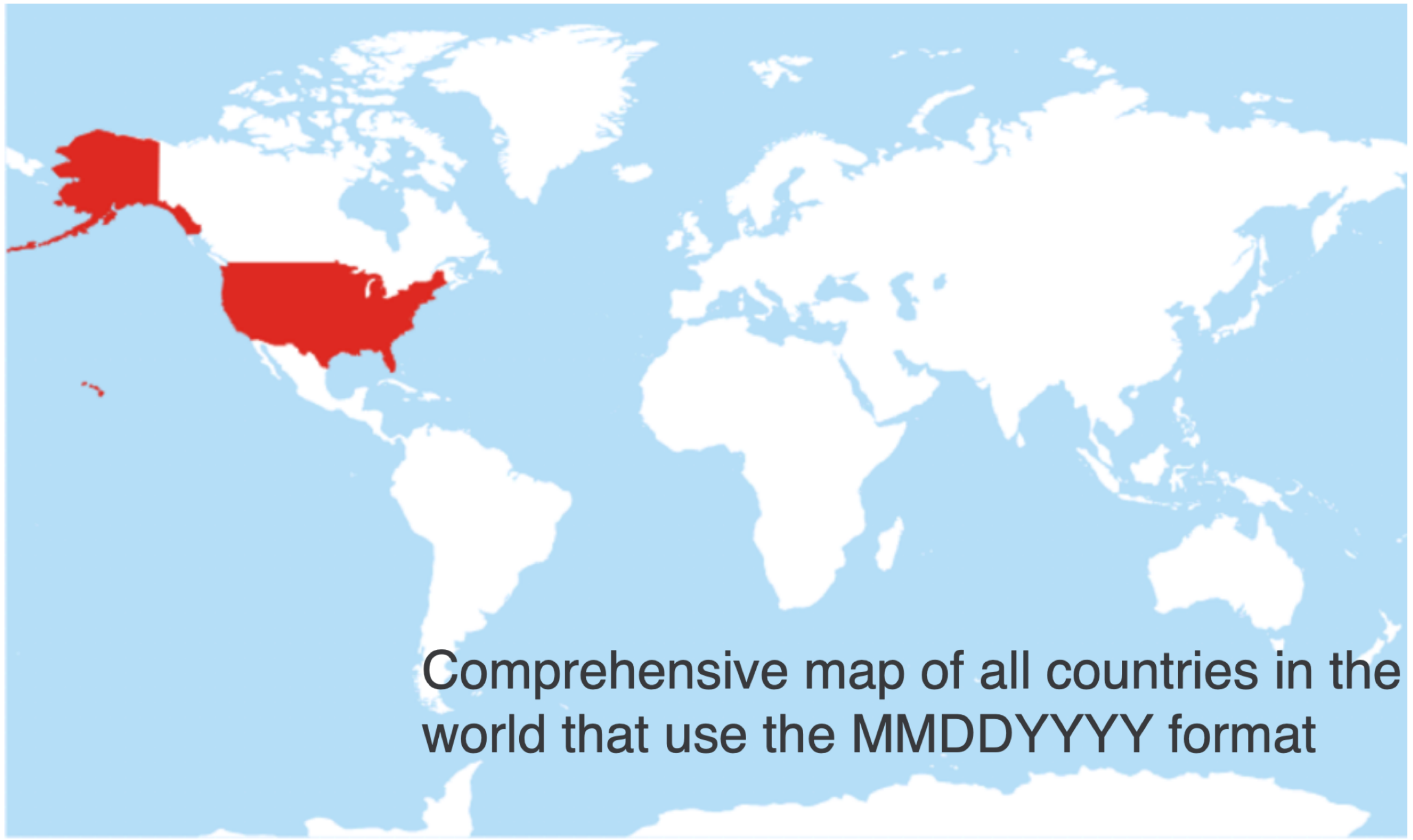
http://www2.stat.duke.edu/~rcs46/lectures_2015/01-markdown-git/slides/naming-slides/naming-slides.pdf

Three principles for (file) names

- 1. Machine readable (do not put special characters and space in the name)
- 2. Human readable (Easy to figure out what the heck something is, based on its name, add slug)
- 3. Plays well with default ordering:
- * Put something numeric first
- * Use the ISO 8601 standard for dates (YYYY-MM-DD)
- * Left pad other numbers with zeros

http://www2.stat.duke.edu/~rcs46/lectures_2015/01-markdown-git/slides/naming-slides/naming-slides.pdf

Jenny Bryan

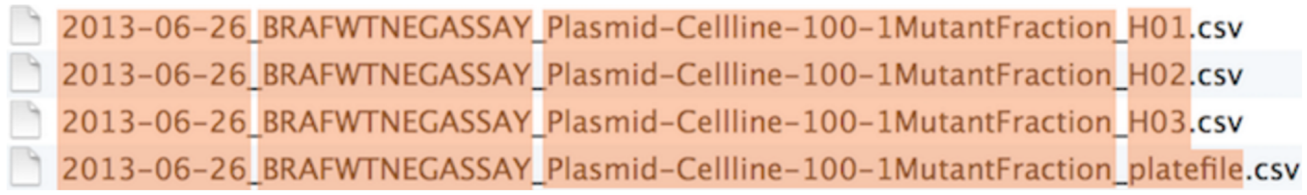

Comprehensive map of all countries in the world that use the MMDDYYYY format

http://www2.stat.duke.edu/~rcs46/lectures_2015/01-markdown-git/slides/naming-slides/naming-slides.pdf

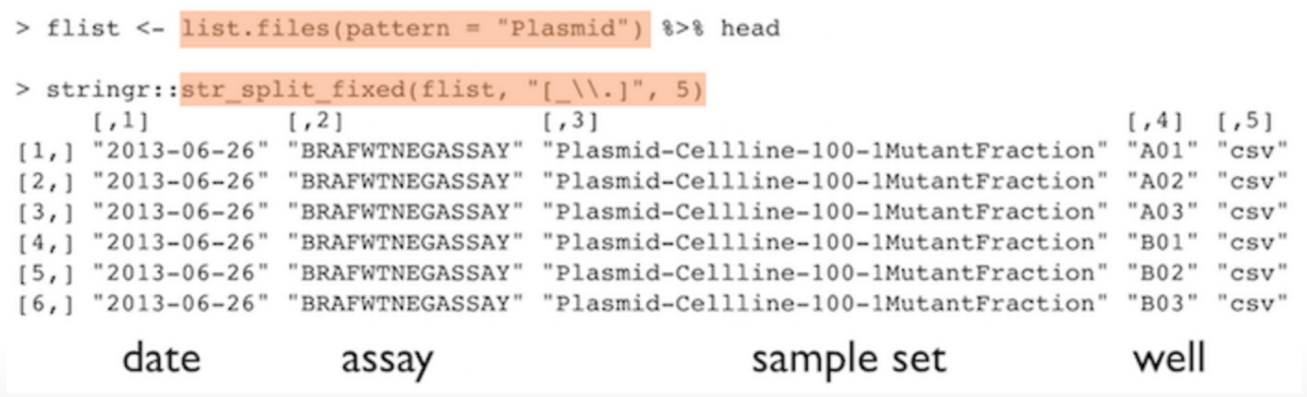
Punctuation

Deliberate use of `"-"` and `"_"` allows recovery of meta-data from the filenames:

- `"_"` underscore used to delimit units of meta-data I want later
- `"-"` hyphen used to delimit words so my eyes don't bleed



```
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H01.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H02.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H03.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_platefile.csv
```

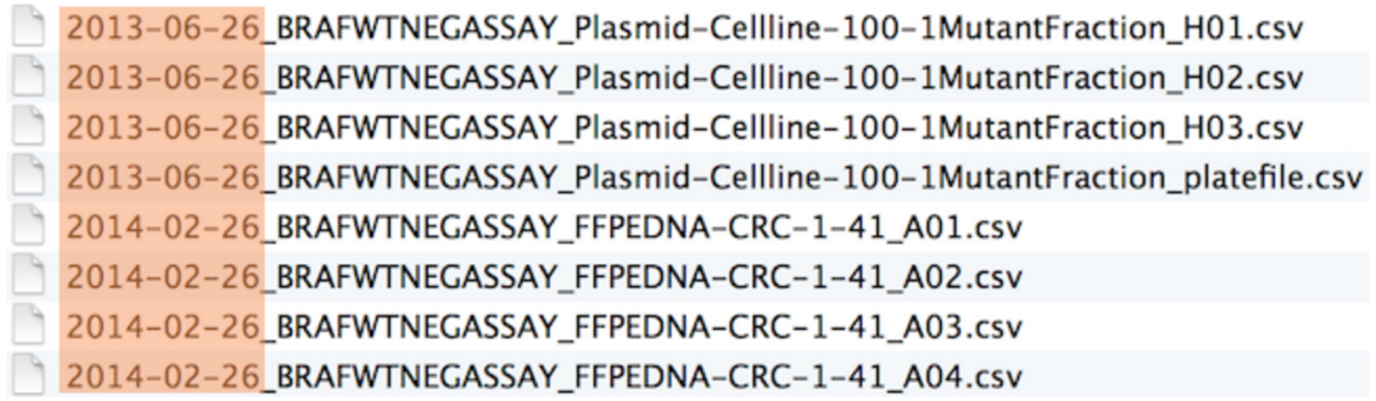


```
> flist <- list.files(pattern = "Plasmid") %>% head

> stringr::str_split_fixed(flist, "[_\\.]", 5)
     [,1]         [,2]             [,3]                                     [,4]  [,5]
[1,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A01" "csv"
[2,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A02" "csv"
[3,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A03" "csv"
[4,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B01" "csv"
[5,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B02" "csv"
[6,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B03" "csv"
```

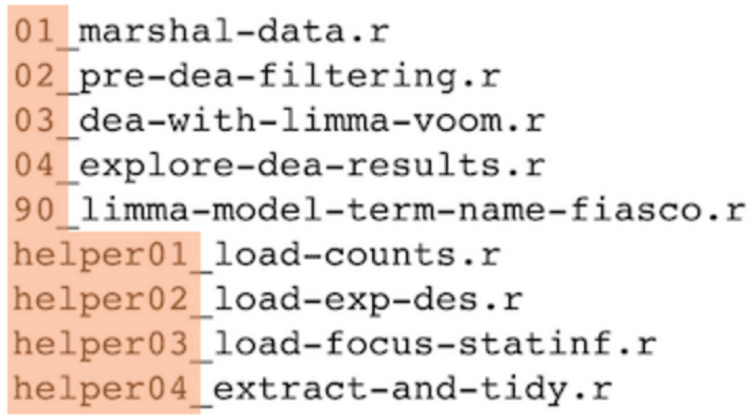
This happens to be R but also possible in the shell, Python, etc.

Jenny Bryan

Go forth and use awesome file names :)



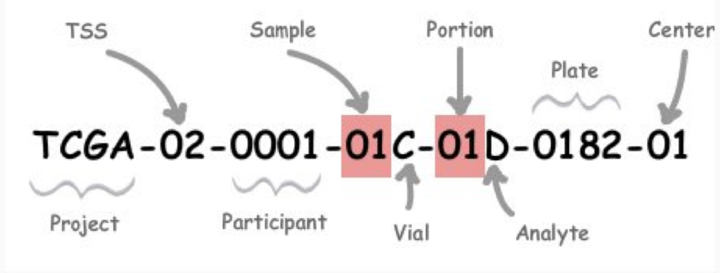
```
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H01.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H02.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H03.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_platefile.csv
2014-02-26_BRAFWTNEGASSAY_FFPEDNA-CRC-1-41_A01.csv
2014-02-26_BRAFWTNEGASSAY_FFPEDNA-CRC-1-41_A02.csv
2014-02-26_BRAFWTNEGASSAY_FFPEDNA-CRC-1-41_A03.csv
2014-02-26_BRAFWTNEGASSAY_FFPEDNA-CRC-1-41_A04.csv
```

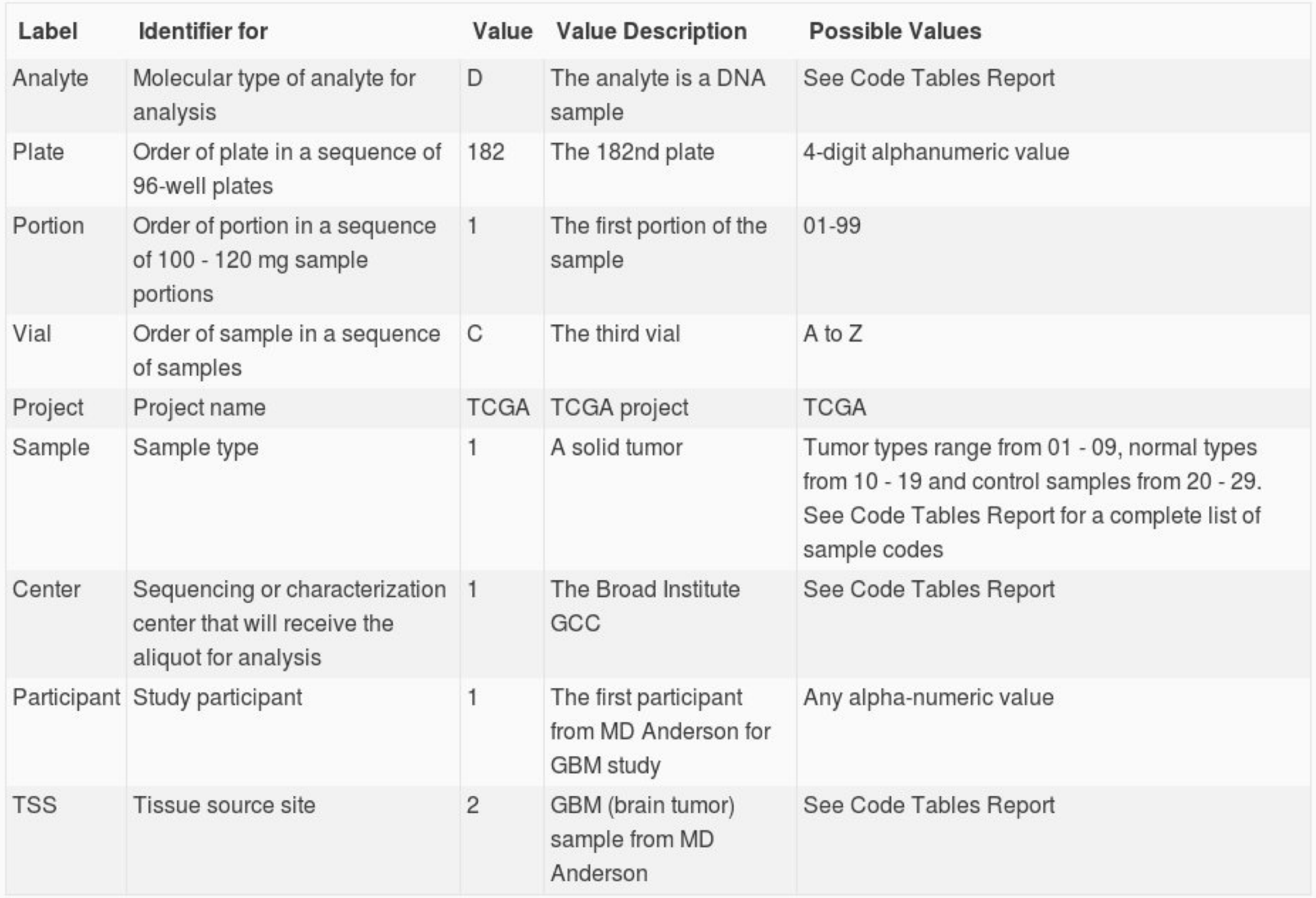


```
01_marshal-data.r
02_pre-dea-filtering.r
03_dea-with-limma-voom.r
04_explore-dea-results.r
90_limma-model-term-name-fiasco.r
helper01_load-counts.r
helper02_load-exp-des.r
helper03_load-focus-statinf.r
helper04_extract-and-tidy.r
```

Jenny Bryan:
https://rawgit.com/Reproducible-Science-Curriculum/rr-organization1/master/organization-01-slides.html

Good example in biology: TCGA barcode





Label	Identifier for	Value	Value Description	Possible Values
Analyte	Molecular type of analyte for analysis	D	The analyte is a DNA sample	See Code Tables Report
Plate	Order of plate in a sequence of 96-well plates	182	The 182nd plate	4-digit alphanumeric value
Portion	Order of portion in a sequence of 100 - 120 mg sample portions	1	The first portion of the sample	01-99
Vial	Order of sample in a sequence of samples	C	The third vial	A to Z
Project	Project name	TCGA	TCGA project	TCGA
Sample	Sample type	1	A solid tumor	Tumor types range from 01 - 09, normal types from 10 - 19 and control samples from 20 - 29. See Code Tables Report for a complete list of sample codes
Center	Sequencing or characterization center that will receive the aliquot for analysis	1	The Broad Institute GCC	See Code Tables Report
Participant	Study participant	1	The first participant from MD Anderson for GBM study	Any alpha-numeric value
TSS	Tissue source site	2	GBM (brain tumor) sample from MD Anderson	See Code Tables Report

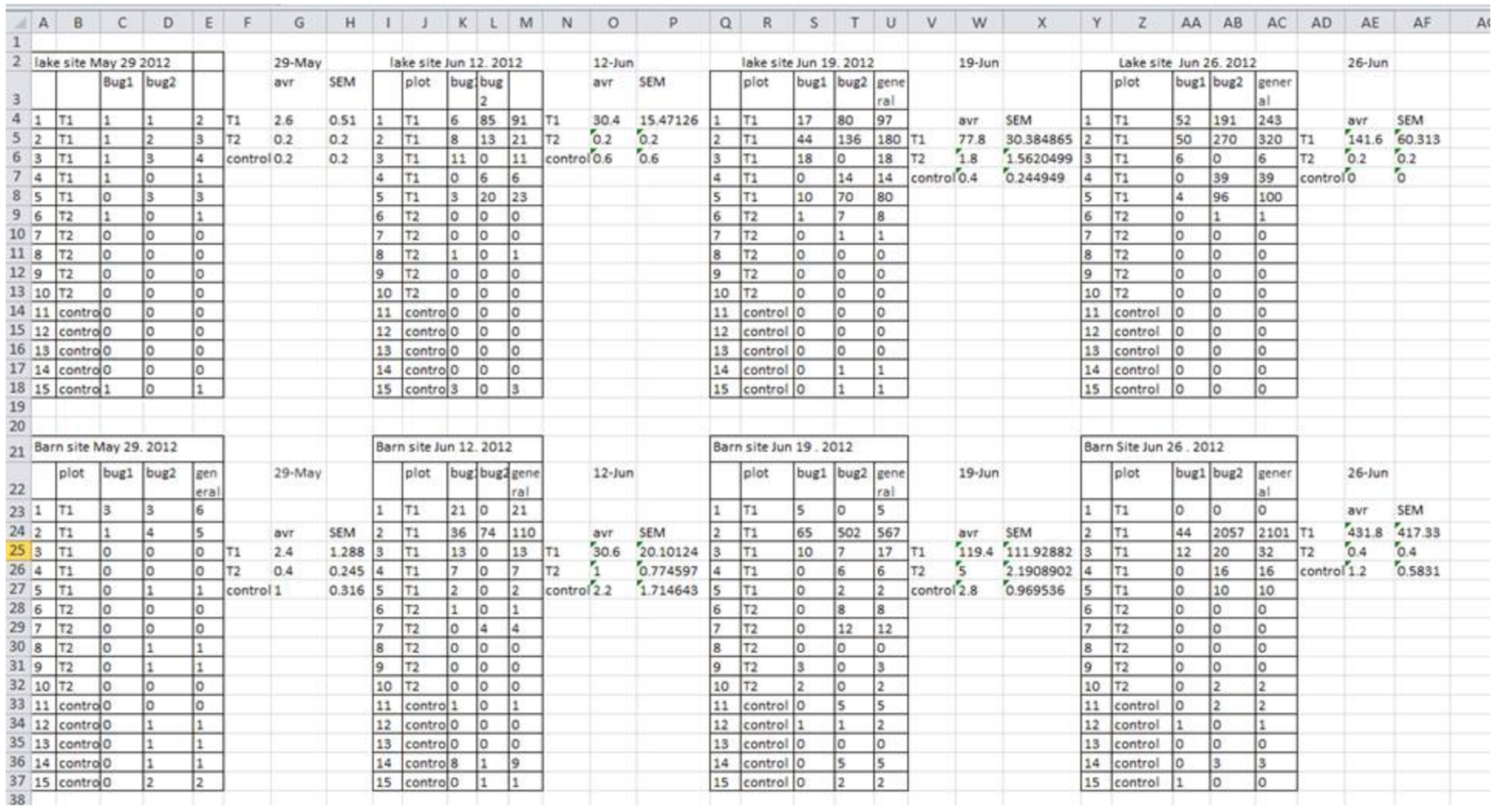
Use excel/spreadsheet wisely

- “There are a few potential errors to be on the lookout for in your own data as well as data from collaborators or the Internet. If you are aware of the errors and the possible negative effect on downstream data analysis and result interpretation, it might motivate yourself and your project members to try and avoid them. **Making small changes to the way you format your data in spreadsheets can have a great impact on efficiency and reliability when it comes to data cleaning and analysis**”.
- https://datacarpentry.org/spreadsheet-ecology-lesson/02-common-mistakes/

Multiple tables in the same sheet

Using multiple tables

A common strategy is creating multiple data tables within one spreadsheet. This confuses the computer, so don't do this! When you create multiple tables within one spreadsheet, you're drawing false associations between things for the computer, which sees each row as an observation. You're also potentially using the same field name in multiple places, which will make it harder to clean your data up into a usable form. The example below depicts the problem:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1																																
2	lake site May 29 2012						29-May		lake site Jun 12. 2012						12-Jun			lake site Jun 19. 2012					19-Jun			Lake site Jun 26. 2012					26-Jun	
3			Bug1	bug2			avr	SEM		plot	bug	bug 2			avr	SEM		plot	bug1	bug2	general					plot	bug1	bug2	general			
4	1	T1	1	1	2	T1	2.6	0.51	1	T1	6	85	91	T1	30.4	15.47126	1	T1	17	80	97		avr	SEM	1	T1	52	191	243		avr	SEM
5	2	T1	1	2	3	T2	0.2	0.2	2	T1	8	13	21	T2	0.2	0.2	2	T1	44	136	180	T1	77.8	30.384865	2	T1	50	270	320	T1	141.6	60.313
6	3	T1	1	3	4	control	0.2	0.2	3	T1	11	0	11	control	0.6	0.6	3	T1	18	0	18	T2	1.8	1.5620499	3	T1	6	0	6	T2	0.2	0.2
7	4	T1	1	0	1				4	T1	0	6	6				4	T1	0	14	14	control	0.4	0.244949	4	T1	0	39	39	control	0	0
8	5	T1	0	3	3				5	T1	3	20	23				5	T1	10	70	80				5	T1	4	96	100			
9	6	T2	1	0	1				6	T2	0	0	0				6	T2	1	7	8				6	T2	0	1	1			
10	7	T2	0	0	0				7	T2	0	0	0				7	T2	0	1	1				7	T2	0	0	0			
11	8	T2	0	0	0				8	T2	1	0	1				8	T2	0	0	0				8	T2	0	0	0			
12	9	T2	0	0	0				9	T2	0	0	0				9	T2	0	0	0				9	T2	0	0	0			
13	10	T2	0	0	0				10	T2	0	0	0				10	T2	0	0	0				10	T2	0	0	0			
14	11	contro	0	0	0				11	contro	0	0	0				11	control	0	0	0				11	control	0	0	0			
15	12	contro	0	0	0				12	contro	0	0	0				12	control	0	0	0				12	control	0	0	0			
16	13	contro	0	0	0				13	contro	0	0	0				13	control	0	0	0				13	control	0	0	0			
17	14	contro	0	0	0				14	contro	0	0	0				14	control	0	1	1				14	control	0	0	0			
18	15	contro	1	0	1				15	contro	3	0	3				15	control	0	1	1				15	control	0	0	0			
19																																
20																																
21	Barn site May 29. 2012								Barn site Jun 12. 2012								Barn site Jun 19 . 2012								Barn Site Jun 26 . 2012							
22		plot	bug1	bug2	general		29-May			plot	bug	bug2	general		12-Jun			plot	bug1	bug2	general		19-Jun			plot	bug1	bug2	general		26-Jun	
23	1	T1	3	3	6				1	T1	21	0	21				1	T1	5	0	5				1	T1	0	0	0		avr	SEM
24	2	T1	1	4	5		avr	SEM	2	T1	36	74	110		avr	SEM	2	T1	65	502	567		avr	SEM	2	T1	44	2057	2101	T1	431.8	417.33
25	3	T1	0	0	0	T1	2.4	1.288	3	T1	13	0	13	T1	30.6	20.10124	3	T1	10	7	17	T1	119.4	111.92882	3	T1	12	20	32	T2	0.4	0.4
26	4	T1	0	0	0	T2	0.4	0.245	4	T1	7	0	7	T2	1	0.774597	4	T1	0	6	6	T2	5	2.1908902	4	T1	0	16	16	control	1.2	0.5831
27	5	T1	0	1	1	control	1	0.316	5	T1	2	0	2	control	2.2	1.714643	5	T1	0	2	2	control	2.8	0.969536	5	T1	0	10	10			
28	6	T2	0	0	0				6	T2	1	0	1				6	T2	0	8	8				6	T2	0	0	0			
29	7	T2	0	0	0				7	T2	0	4	4				7	T2	0	12	12				7	T2	0	0	0			
30	8	T2	0	1	1				8	T2	0	0	0				8	T2	0	0	0				8	T2	0	0	0			
31	9	T2	0	1	1				9	T2	0	0	0				9	T2	3	0	3				9	T2	0	0	0			
32	10	T2	0	0	0				10	T2	0	0	0				10	T2	2	0	2				10	T2	0	2	2			
33	11	contro	0	0	0				11	contro	1	0	1				11	control	0	5	5				11	control	0	2	2			
34	12	contro	0	1	1				12	contro	0	0	0				12	control	1	1	2				12	control	1	0	1			
35	13	contro	0	1	1				13	contro	0	0	0				13	control	0	0	0				13	control	0	0	0			
36	14	contro	0	1	1				14	contro	8	1	9				14	control	0	5	5				14	control	0	3	3			
37	15	contro	0	2	2				15	contro	0	1	1				15	control	0	2	2				15	control	1	0	0			
38																																

Using problematic null values

Example: using -999 or other numerical values (or zero) to represent missing data.

Solutions:

There are a few reasons why null values get represented differently within a dataset. Sometimes confusing null values are automatically recorded from the measuring device. If that's the case, there's not much you can do, but it can be addressed in data cleaning with a tool like OpenRefine before analysis. Other times different null values are used to convey different reasons why the data isn't there. This is important information to capture, but is in effect using one column to capture two pieces of information. Like for using formatting to convey information it would be good here to create a new column like 'data_missing' and use that column to capture the different reasons.

Whatever the reason, it's a problem if unknown or missing data is recorded as -999, 999, or 0. Many statistical programs will not recognize that these are intended to represent missing (null) values. How these values are interpreted will depend on the software you use to analyze your data. It is essential to use a clearly defined and consistent null indicator. Blanks (most applications) and NA (for R) are good choices. White et al, 2013, explain good choices for indicating null values for different software applications in their article: Nine simple ways to make it easier to (re)use your data. Ideas in Ecology and Evolution.

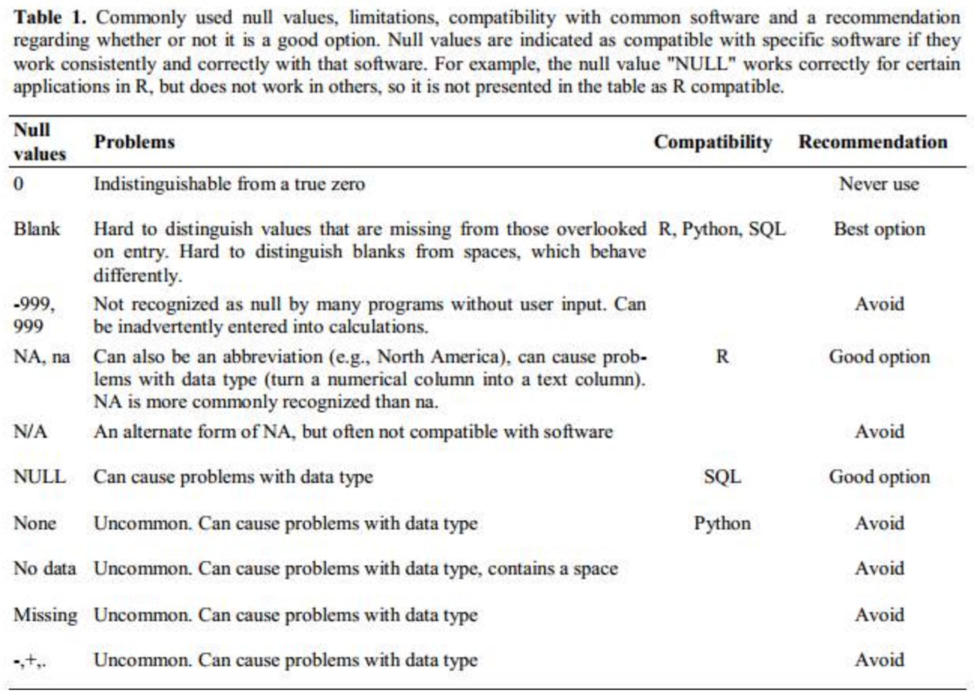
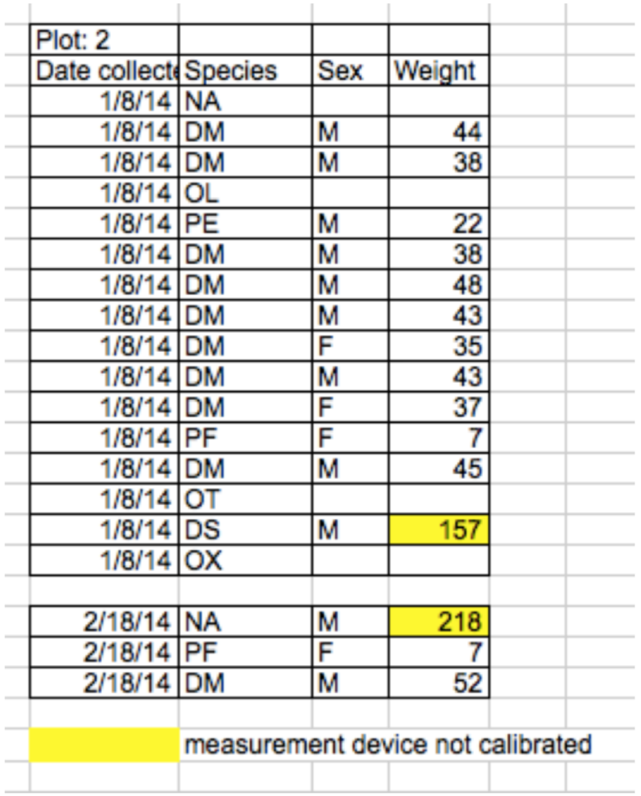


Table 1. Commonly used null values, limitations, compatibility with common software and a recommendation regarding whether or not it is a good option. Null values are indicated as compatible with specific software if they work consistently and correctly with that software. For example, the null value "NULL" works correctly for certain applications in R, but does not work in others, so it is not presented in the table as R compatible.

Null values	Problems	Compatibility	Recommendation
0	Indistinguishable from a true zero		Never use
Blank	Hard to distinguish values that are missing from those overlooked on entry. Hard to distinguish blanks from spaces, which behave differently.	R, Python, SQL	Best option
-999, 999	Not recognized as null by many programs without user input. Can be inadvertently entered into calculations.		Avoid
NA, na	Can also be an abbreviation (e.g., North America), can cause problems with data type (turn a numerical column into a text column). NA is more commonly recognized than na.	R	Good option
N/A	An alternate form of NA, but often not compatible with software		Avoid
NULL	Can cause problems with data type	SQL	Good option
None	Uncommon. Can cause problems with data type	Python	Avoid
No data	Uncommon. Can cause problems with data type, contains a space		Avoid
Missing	Uncommon. Can cause problems with data type		Avoid
-,+,.	Uncommon. Can cause problems with data type		Avoid

Using formatting to convey information

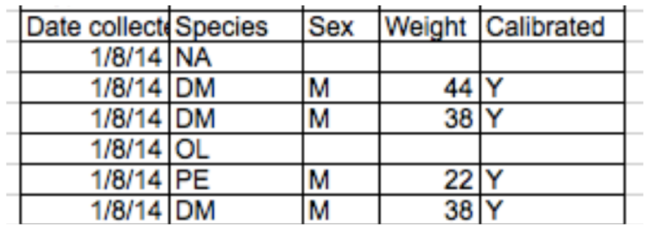
Example: highlighting cells, rows or columns that should be excluded from an analysis, leaving blank rows to indicate separations in data.



Plot: 2			
Date collecte	Species	Sex	Weight
1/8/14	NA		
1/8/14	DM	M	44
1/8/14	DM	M	38
1/8/14	OL		
1/8/14	PE	M	22
1/8/14	DM	M	38
1/8/14	DM	M	48
1/8/14	DM	M	43
1/8/14	DM	F	35
1/8/14	DM	M	43
1/8/14	DM	F	37
1/8/14	PF	F	7
1/8/14	DM	M	45
1/8/14	OT		
1/8/14	DS	M	157
1/8/14	OX		
2/18/14	NA	M	218
2/18/14	PF	F	7
2/18/14	DM	M	52

🟨 measurement device not calibrated

Solution: create a new field to encode which data should be excluded.



Date collecte	Species	Sex	Weight	Calibrated
1/8/14	NA			
1/8/14	DM	M	44	Y
1/8/14	DM	M	38	Y
1/8/14	OL			
1/8/14	PE	M	22	Y
1/8/14	DM	M	38	Y

Using problematic field names

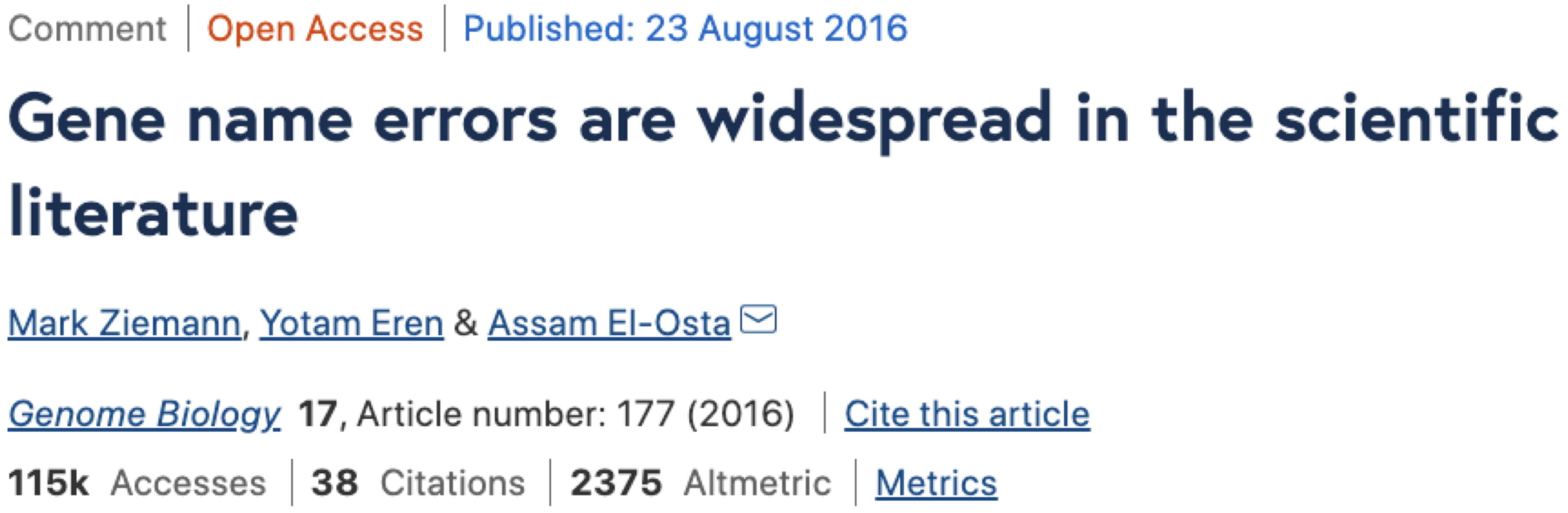
Choose descriptive field names, but be careful not to include spaces, numbers, or special characters of any kind. Spaces can be misinterpreted by parsers that use whitespace as delimiters and some programs don’t like field names that are text strings that start with numbers.

Underscores (`_`) are a good alternative to spaces. Consider writing names in camel case (like this: ExampleFileName) to improve readability. Remember that abbreviations that make sense at the moment may not be so obvious in 6 months, but don’t overdo it with names that are excessively long. Including the units in the field names avoids confusion and enables others to readily interpret your fields.

Examples

Good Name	Good Alternative	Avoid
Max_temp_C	MaxTemp	Maximum Temp (°C)
Precipitation_mm	Precipitation	precmm
Mean_year_growth	MeanYearGrowth	Mean growth/year
sex	sex	M/F
weight	weight	w.
cell_type	CellType	Cell Type
Observation_01	first_observation	1st Obs

Be cautious with excel



Comment | Open Access | Published: 23 August 2016

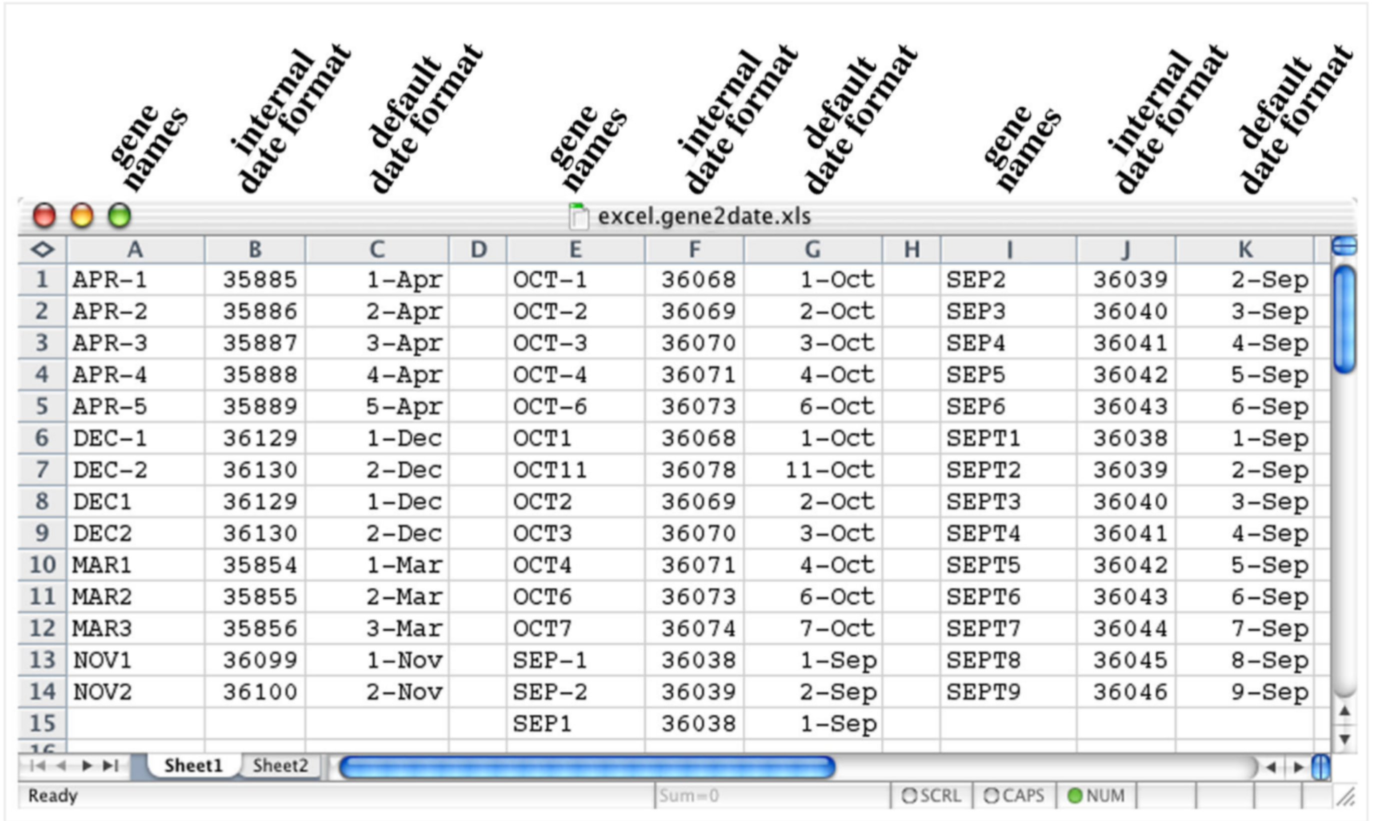
Gene name errors are widespread in the scientific literature

Mark Ziemann, Yotam Eren & Assam El-Osta

Genome Biology **17**, Article number: 177 (2016) | Cite this article

115k Accesses | **38** Citations | **2375** Altmetric | Metrics

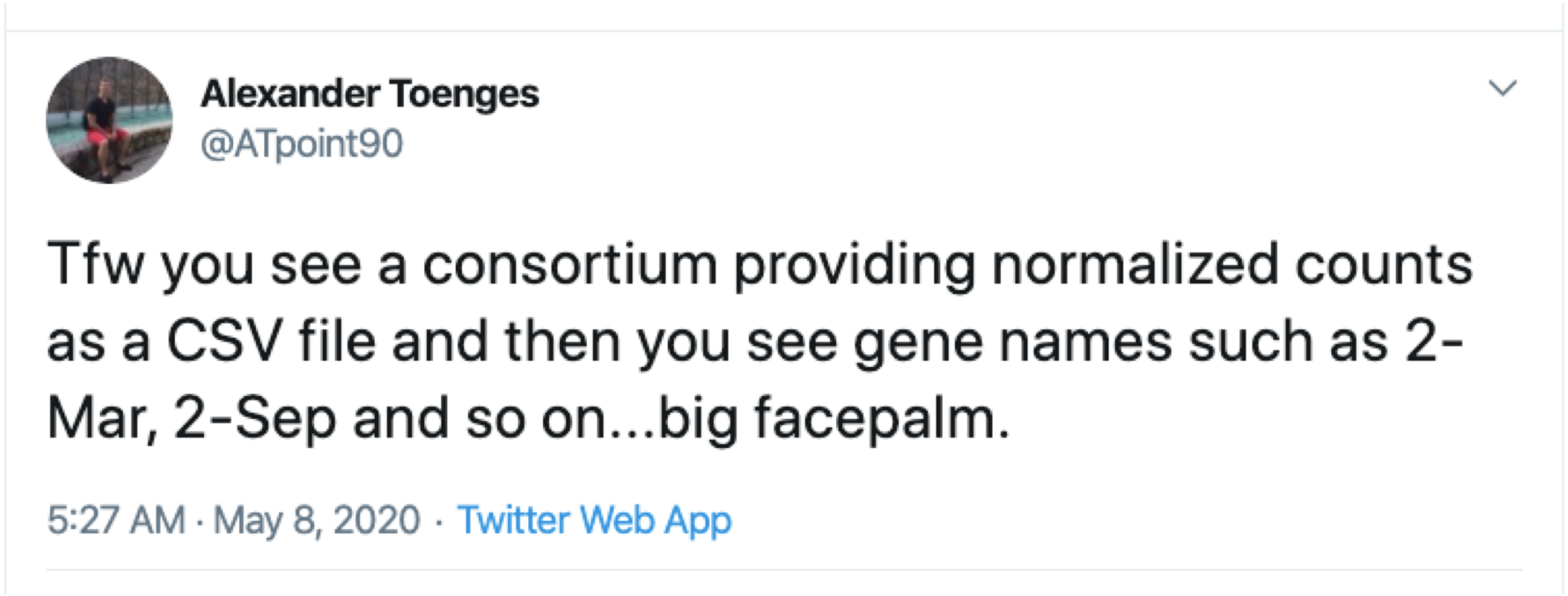
OCT4!



	A	B	C	D	E	F	G	H	I	J	K
	gene names	internal date format	default date format		gene names	internal date format	default date format		gene names	internal date format	default date format
1	APR-1	35885	1-Apr		OCT-1	36068	1-Oct		SEP2	36039	2-Sep
2	APR-2	35886	2-Apr		OCT-2	36069	2-Oct		SEP3	36040	3-Sep
3	APR-3	35887	3-Apr		OCT-3	36070	3-Oct		SEP4	36041	4-Sep
4	APR-4	35888	4-Apr		OCT-4	36071	4-Oct		SEP5	36042	5-Sep
5	APR-5	35889	5-Apr		OCT-6	36073	6-Oct		SEP6	36043	6-Sep
6	DEC-1	36129	1-Dec		OCT1	36068	1-Oct		SEPT1	36038	1-Sep
7	DEC-2	36130	2-Dec		OCT11	36078	11-Oct		SEPT2	36039	2-Sep
8	DEC1	36129	1-Dec		OCT2	36069	2-Oct		SEPT3	36040	3-Sep
9	DEC2	36130	2-Dec		OCT3	36070	3-Oct		SEPT4	36041	4-Sep
10	MAR1	35854	1-Mar		OCT4	36071	4-Oct		SEPT5	36042	5-Sep
11	MAR2	35855	2-Mar		OCT6	36073	6-Oct		SEPT6	36043	6-Sep
12	MAR3	35856	3-Mar		OCT7	36074	7-Oct		SEPT7	36044	7-Sep
13	NOV1	36099	1-Nov		SEP-1	36038	1-Sep		SEPT8	36045	8-Sep
14	NOV2	36100	2-Nov		SEP-2	36039	2-Sep		SEPT9	36046	9-Sep
15					SEP1	36038	1-Sep				

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-5-80

It is still not uncommon nowadays

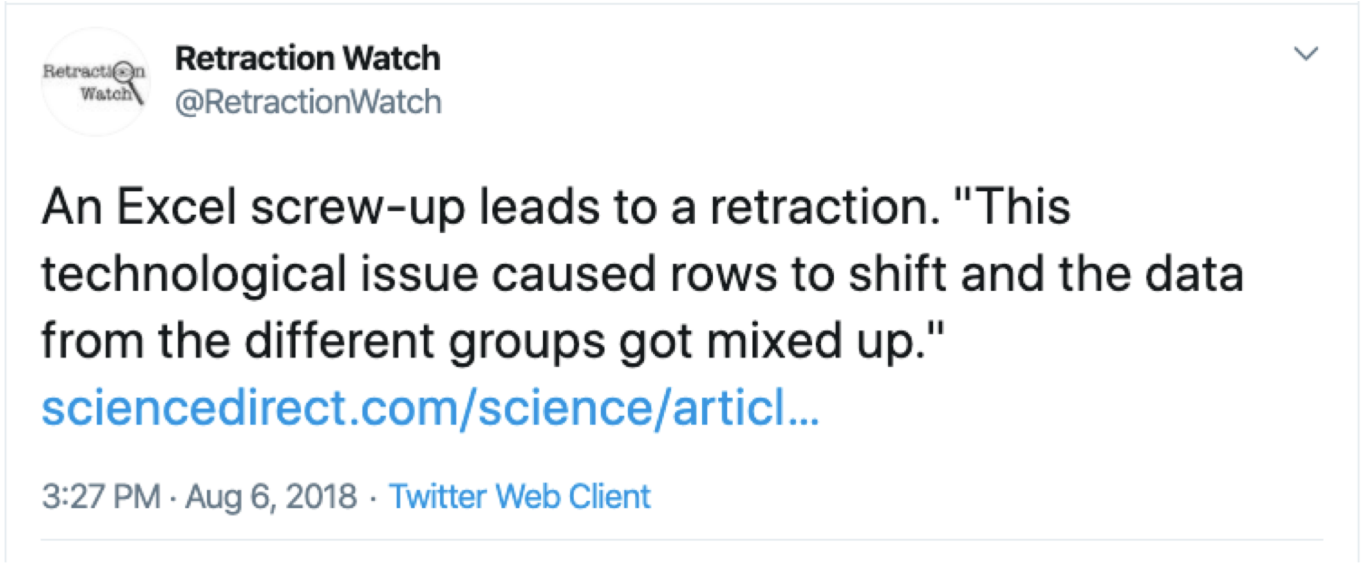


Alexander Toenges
@ATpoint90

Tfw you see a consortium providing normalized counts as a CSV file and then you see gene names such as 2-Mar, 2-Sep and so on...big facepalm.

5:27 AM · May 8, 2020 · Twitter Web App

Excel misusage can lead to retraction


Retraction Watch
@RetractionWatch

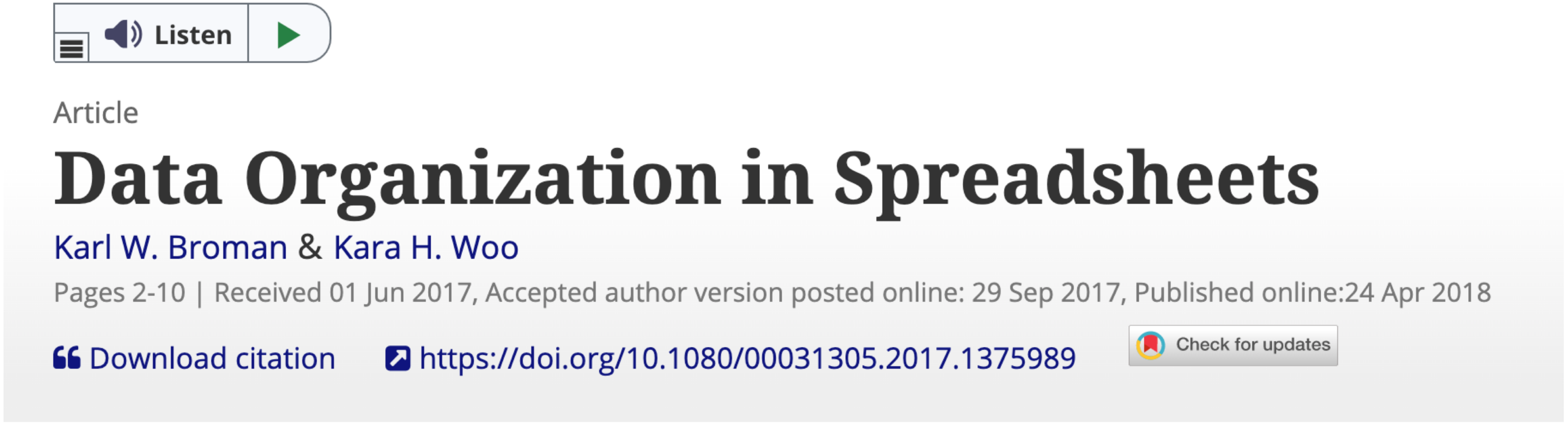
An Excel screw-up leads to a retraction. "This technological issue caused rows to shift and the data from the different groups got mixed up." sciencedirect.com/science/articl...

3:27 PM · Aug 6, 2018 · Twitter Web Client

https://www.sciencedirect.com/science/article/pii/S0018506X18302599?via%3Dihub

https://github.com/jennybc/scary-excel-stories by Jenny Bryan

Recommended reading


Listen

Article

Data Organization in Spreadsheets

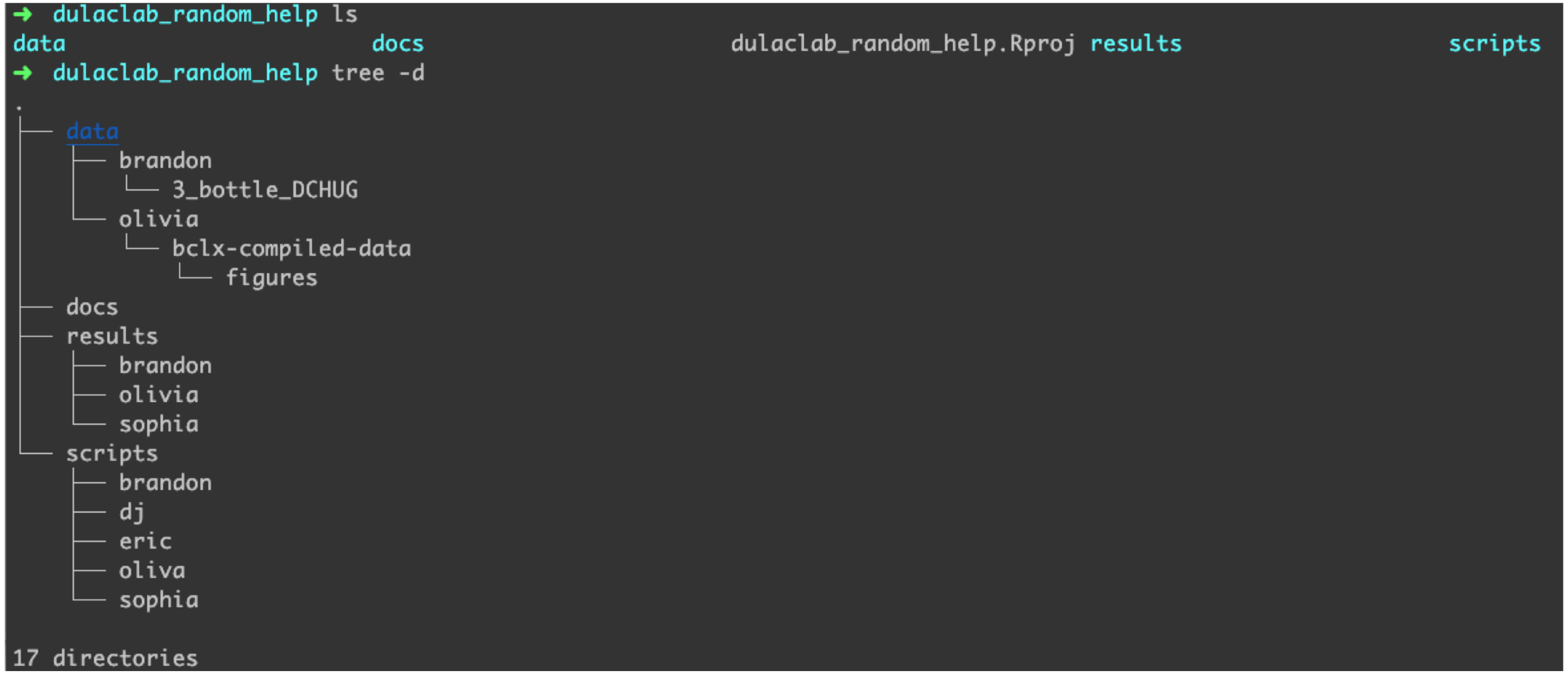
Karl W. Broman & Kara H. Woo

Pages 2-10 | Received 01 Jun 2017, Accepted author version posted online: 29 Sep 2017, Published online:24 Apr 2018

Download citation https://doi.org/10.1080/00031305.2017.1375989 Check for updates

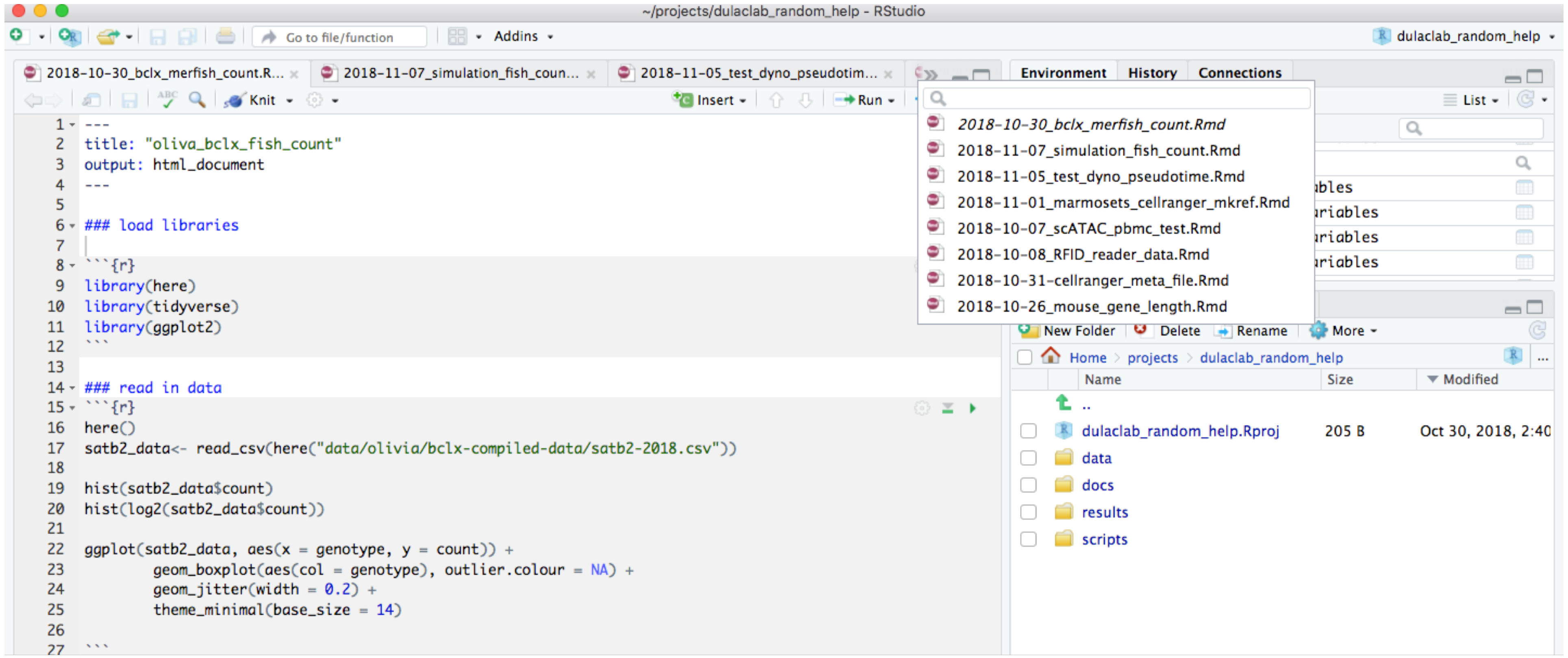
https://www.tandfonline.com/doi/full/10.1080/00031305.2017.1375989

Organization of each project down-stream analysis



```
➜ dulaclab_random_help ls
data                     docs                     dulaclab_random_help.Rproj results                  scripts
➜ dulaclab_random_help tree -d
.
├── data
│   ├── brandon
│   │   └── 3_bottle_DCHUG
│   └── olivia
│       └── bclx-compiled-data
│           └── figures
├── docs
├── results
│   ├── brandon
│   ├── olivia
│   └── sophia
└── scripts
    ├── brandon
    ├── dj
    ├── eric
    ├── oliva
    └── sophia

17 directories
```

Rstudio R project



~~~
---
title: "oliva_bclx_fish_count"
output: html_document
---

### load libraries

```{r}
library(here)
library(tidyverse)
library(ggplot2)
~~~

read in data
```{r}
here()
satb2_data<- read_csv(here("data/olivia/bclx-compiled-data/satb2-2018.csv"))

hist(satb2_data$count)
hist(log2(satb2_data$count))

ggplot(satb2_data, aes(x = genotype, y = count)) +
        geom_boxplot(aes(col = genotype), outlier.colour = NA) +
        geom_jitter(width = 0.2) +
        theme_minimal(base_size = 14)

```
```

2018-10-30_bclx_merfish_count.Rmd
2018-11-07_simulation_fish_count.Rmd
2018-11-05_test_dyno_pseudotime.Rmd
2018-11-01_marmosets_cellranger_mkref.Rmd
2018-10-07_scATAC_pbmc_test.Rmd
2018-10-08_RFID_reader_data.Rmd
2018-10-31-cellranger_meta_file.Rmd
2018-10-26_mouse_gene_length.Rmd

Name	Size	Modified
..		
dulaclab_random_help.Rproj	205 B	Oct 30, 2018, 2:40
data		
docs		
results		
scripts		
```

PLOS | COMPUTATIONAL BIOLOGY

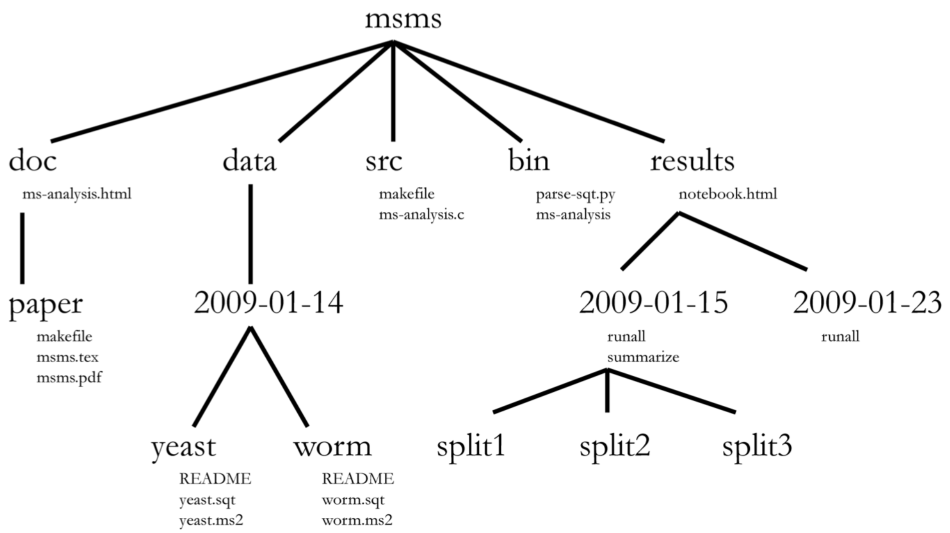
OPEN ACCESS

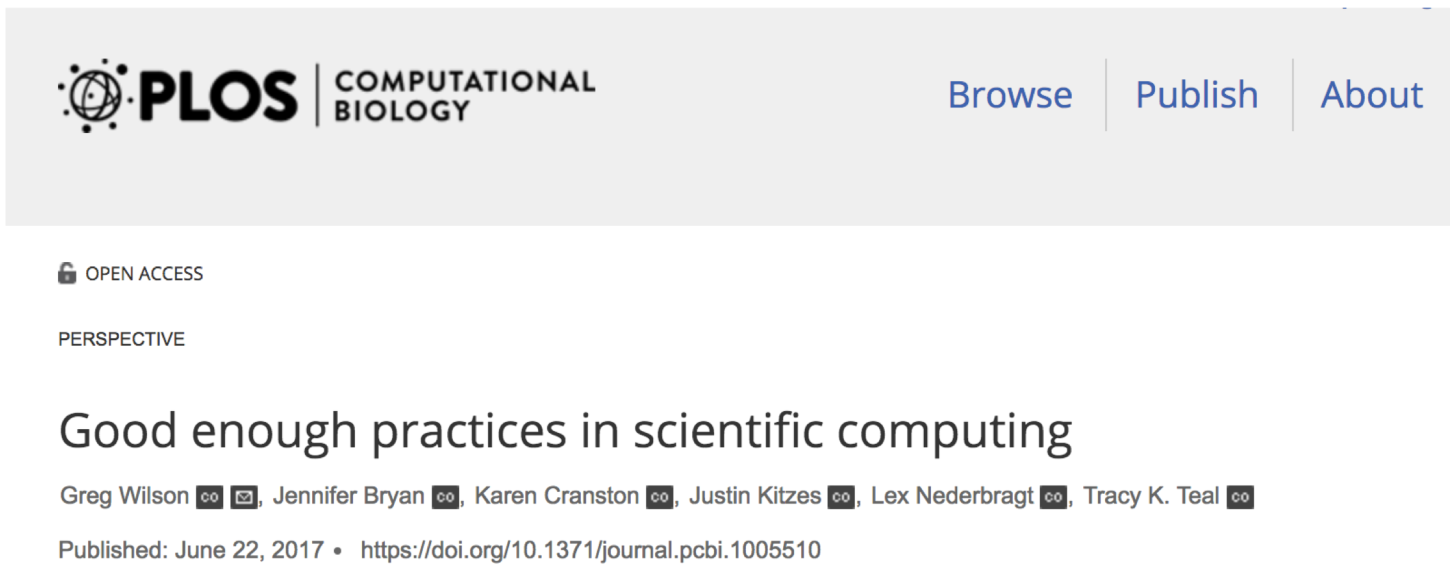
EDUCATION

# A Quick Guide to Organizing Computational Biology Projects

William Stafford Noble

Published: July 31, 2009 • https://doi.org/10.1371/journal.pcbi.1000424





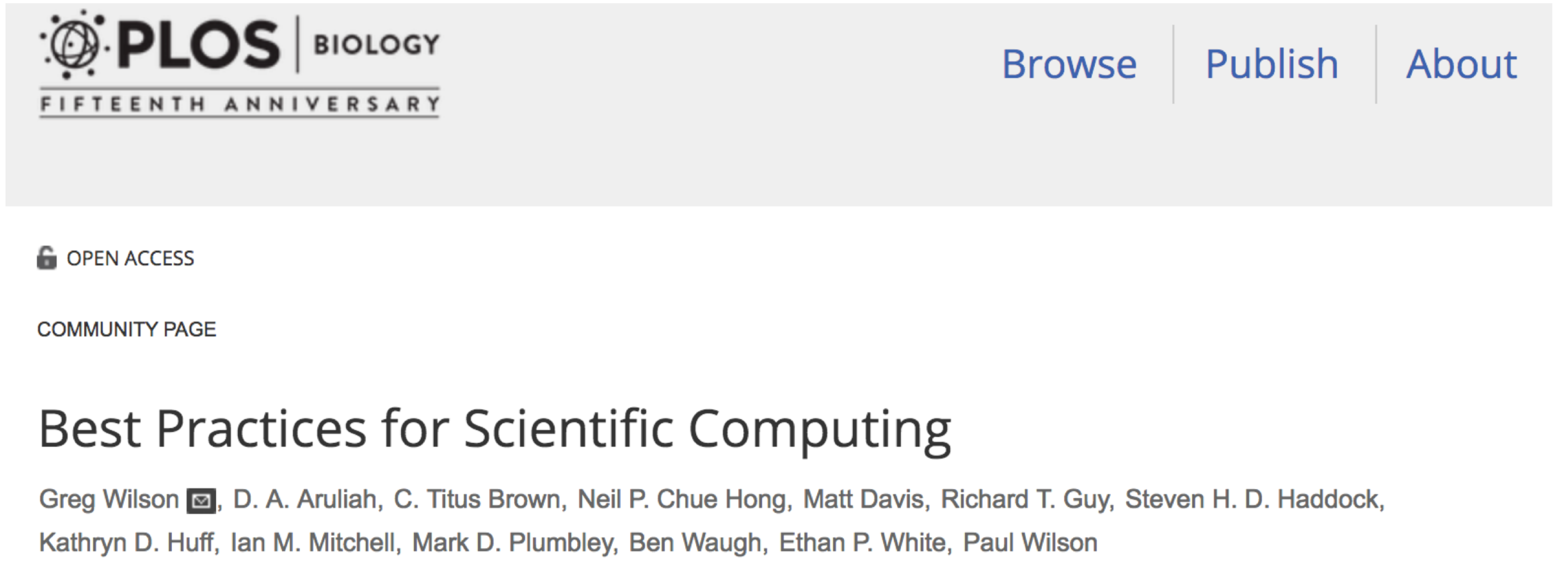
OPEN ACCESS

PERSPECTIVE

Good enough practices in scientific computing

Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, Tracy K. Teal

Published: June 22, 2017 • https://doi.org/10.1371/journal.pcbi.1005510



OPEN ACCESS

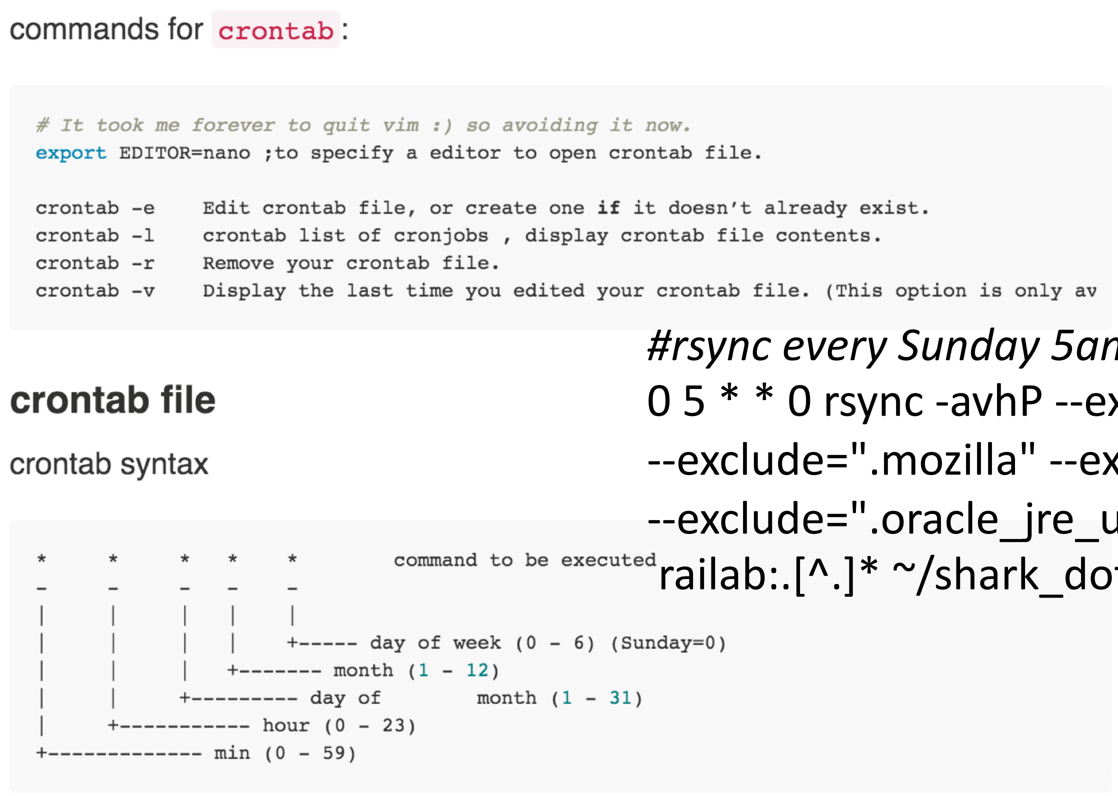
COMMUNITY PAGE

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

# One last suggestion: backup!
# Backup by crontab

- [https://divingintogeneticsandgenomics.rbind.io/post/crontab-for-backup/](https://divingintogeneticsandgenomics.rbind.io/post/crontab-for-backup/)



commands for `crontab`:

```
# It took me forever to quit vim :) so avoiding it now.
export EDITOR=nano ;to specify a editor to open crontab file.

crontab -e    Edit crontab file, or create one if it doesn't already exist.
crontab -l    crontab list of cronjobs , display crontab file contents.
crontab -r    Remove your crontab file.
crontab -v    Display the last time you edited your crontab file. (This option is only av
```

crontab file

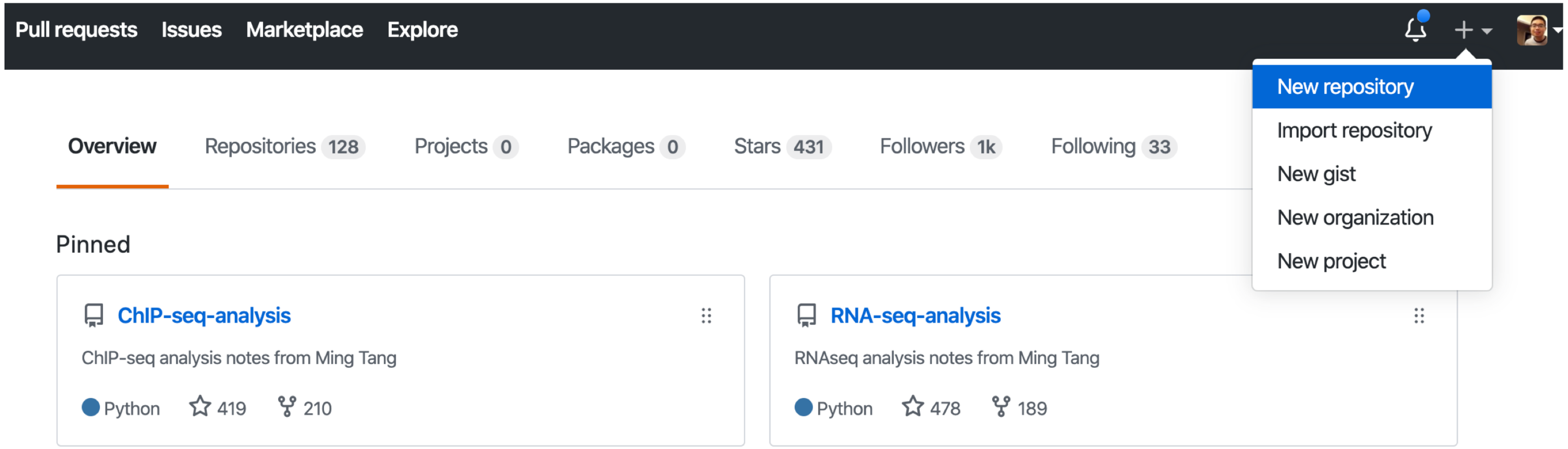
crontab syntax

```
*     *     *   *    *        command to be executed
-     -     -   -    -
|     |     |   |    |
|     |     |   |    +----- day of week (0 - 6) (Sunday=0)
|     |     |   +------- month (1 - 12)
|     |     +--------- day of        month (1 - 31)
|     +----------- hour (0 - 23)
+------------- min (0 - 59)
```

```
#rsync every Sunday 5am.
0 5 * * 0 rsync -avhP --exclude=".aspera" --exclude=".autojump" --exclude=".bash_history"
--exclude=".mozilla" --exclude=".myconfigs"
--exclude=".oracle_jre_usage" --exclude=".parallel" --exclude=".pki" --exclude=".rbenv"
railab:.[^.]* ~/shark_dotfiles >> /var/log/rsync_shark_dotfiles.log 2>&1
```

Reproducible computing using Rstudio: A walk through

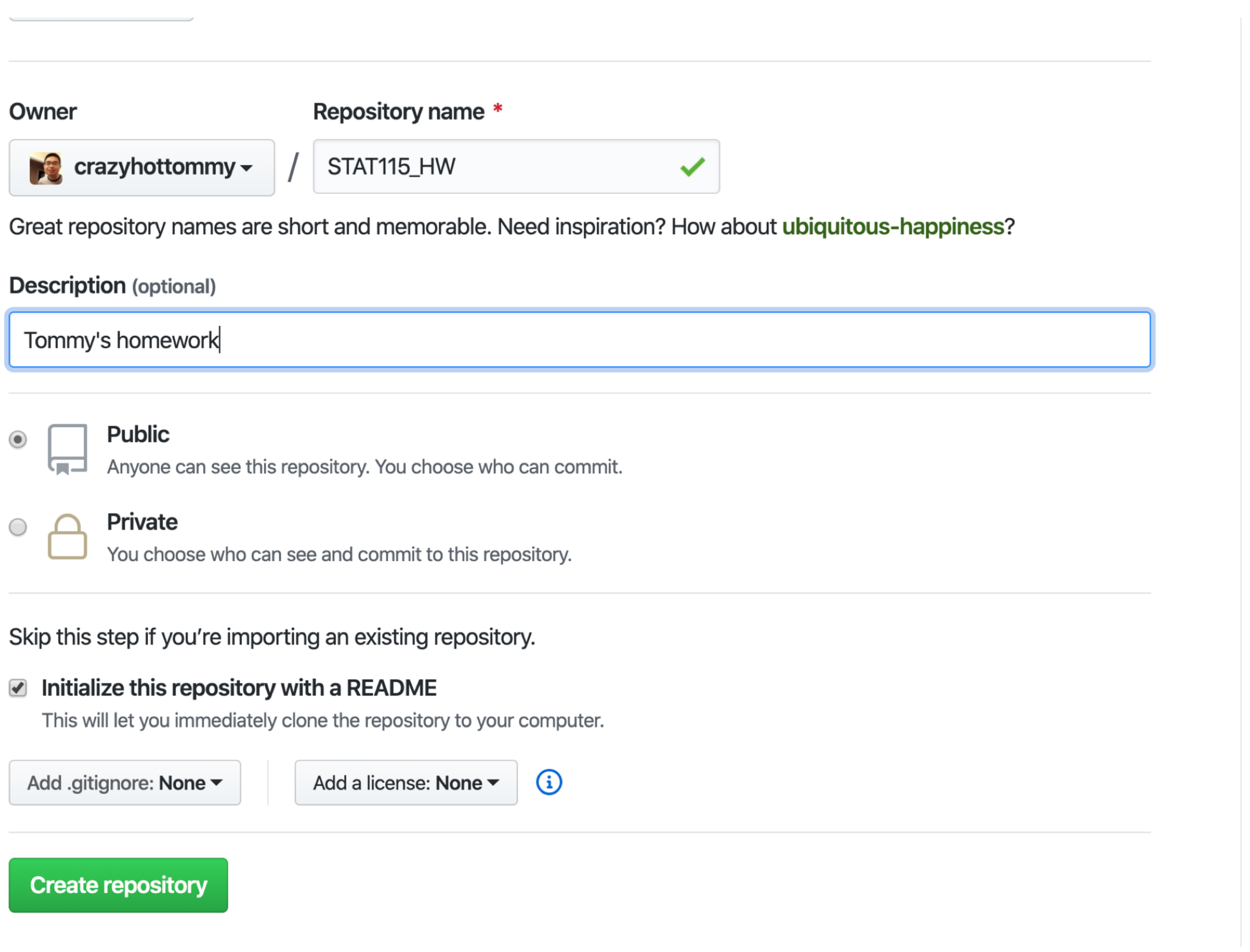
- Go to https://github.com/username
- Create a new repository



ChIP-seq-analysis
ChIP-seq analysis notes from Ming Tang
Python 419 210

RNA-seq-analysis
RNAseq analysis notes from Ming Tang
Python 478 189

Create the new repository
Check [] Initialize this repository with a README



Owner: crazyhottommy / Repository name *: STAT115_HW ✓

Great repository names are short and memorable. Need inspiration? How about ubiquitous-happiness?

Description (optional): Tommy's homework

○ **Public** Anyone can see this repository. You choose who can commit. [✓]

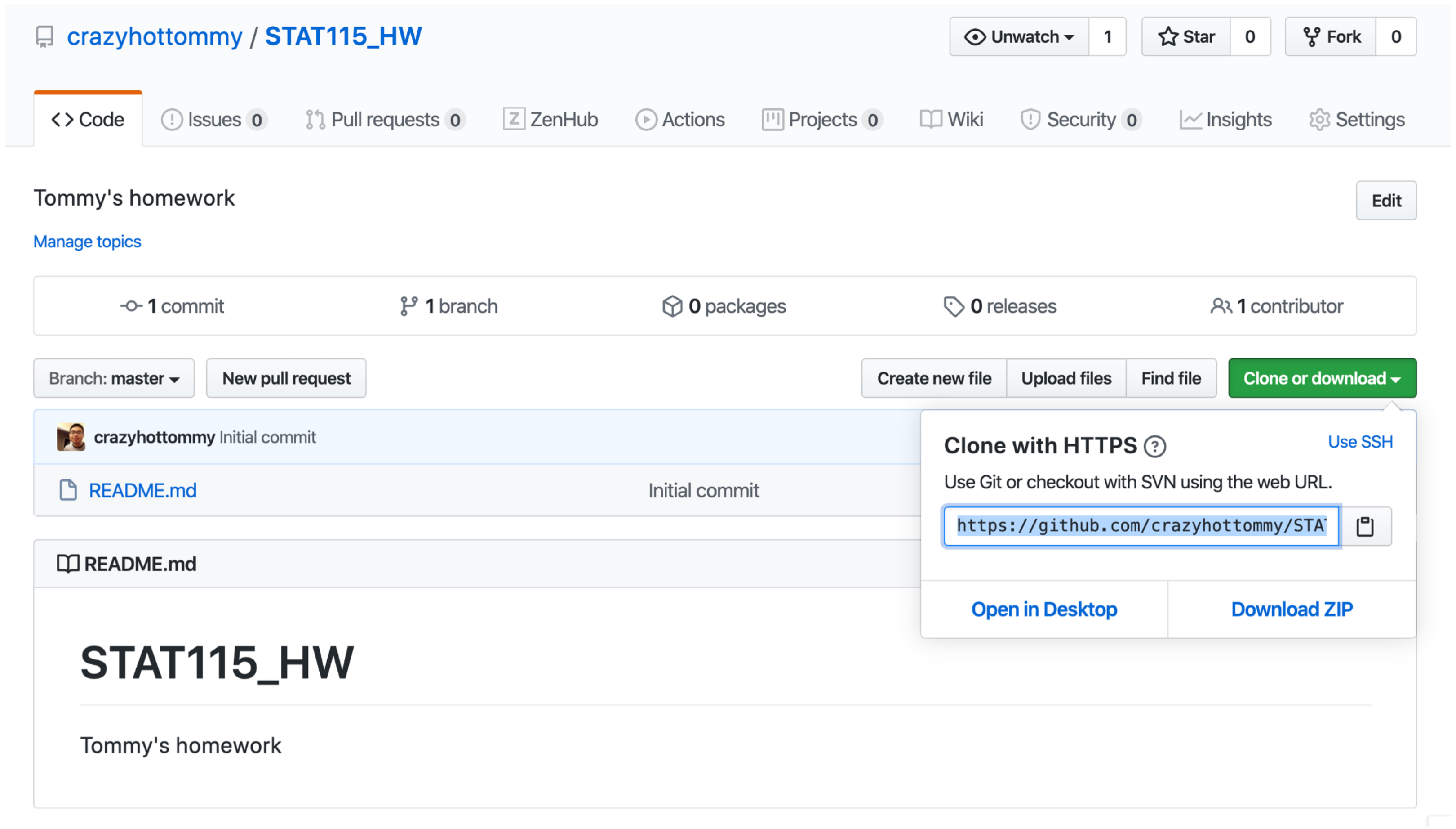
○ **Private** You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

[✓] **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** | Add a license: **None**

Copy the link from “Clone with HTTPS”


crazyhottommy / STAT115_HW

Tommy's homework

crazyhottommy Initial commit

README.md Initial commit

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/crazyhottommy/STA
```

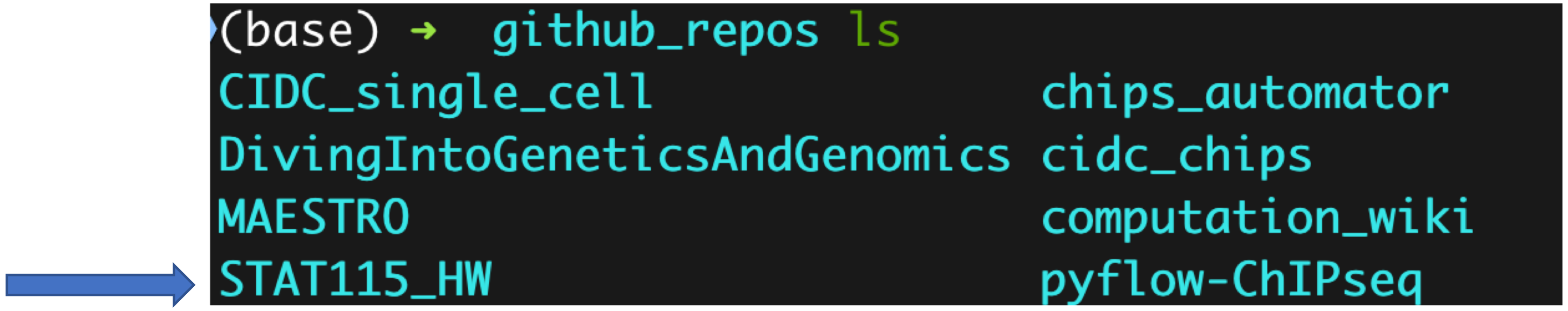
STAT115_HW

Tommy's homework

Go back to your local computer, open terminal

- $ cd /Users/mtang/Dropbox (Partners HealthCare)
- $ mkdir github_repos; cd github_repos
- $ git clone https://github.com/crazyhottommy/STAT115_HW.git
- You should see STAT115_HW folder in the github_repos folder.

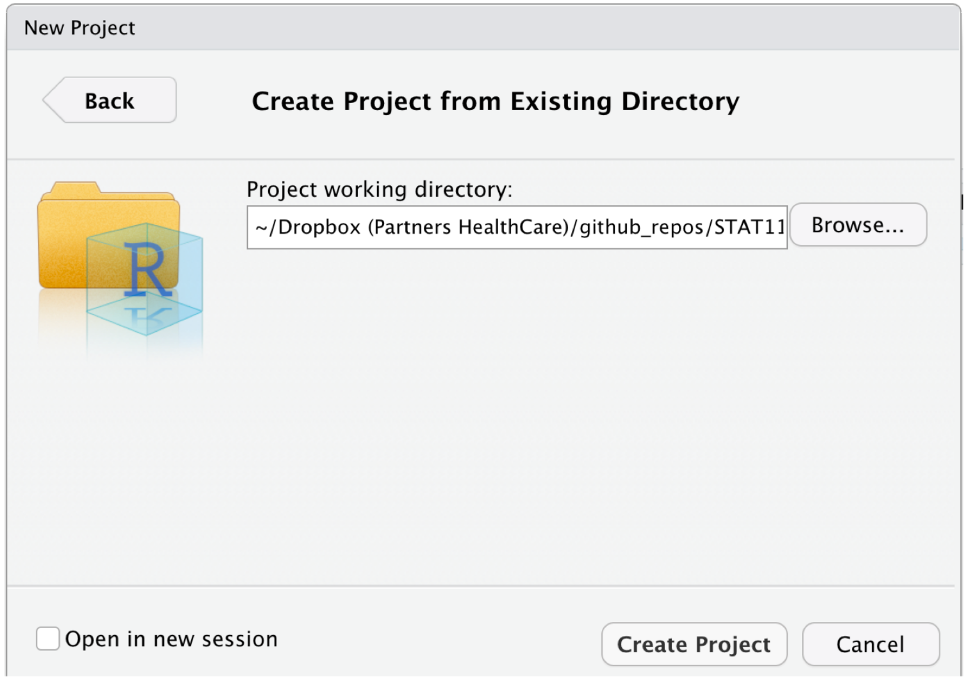
e.g.,



```
(base) ➜  github_repos ls
CIDC_single_cell                chips_automator
DivingIntoGeneticsAndGenomics   cidc_chips
MAESTRO                         computation_wiki
STAT115_HW                      pyflow-ChIPseq
```

I put it in the Dropbox folder since we have unlimited space with Partner's email, so it get backed up in dropbox as well!

Open Rstudio -- > File -- > New Project --> Existing Directory -- > Browse and select the STAT115_HW folder --> Create Project



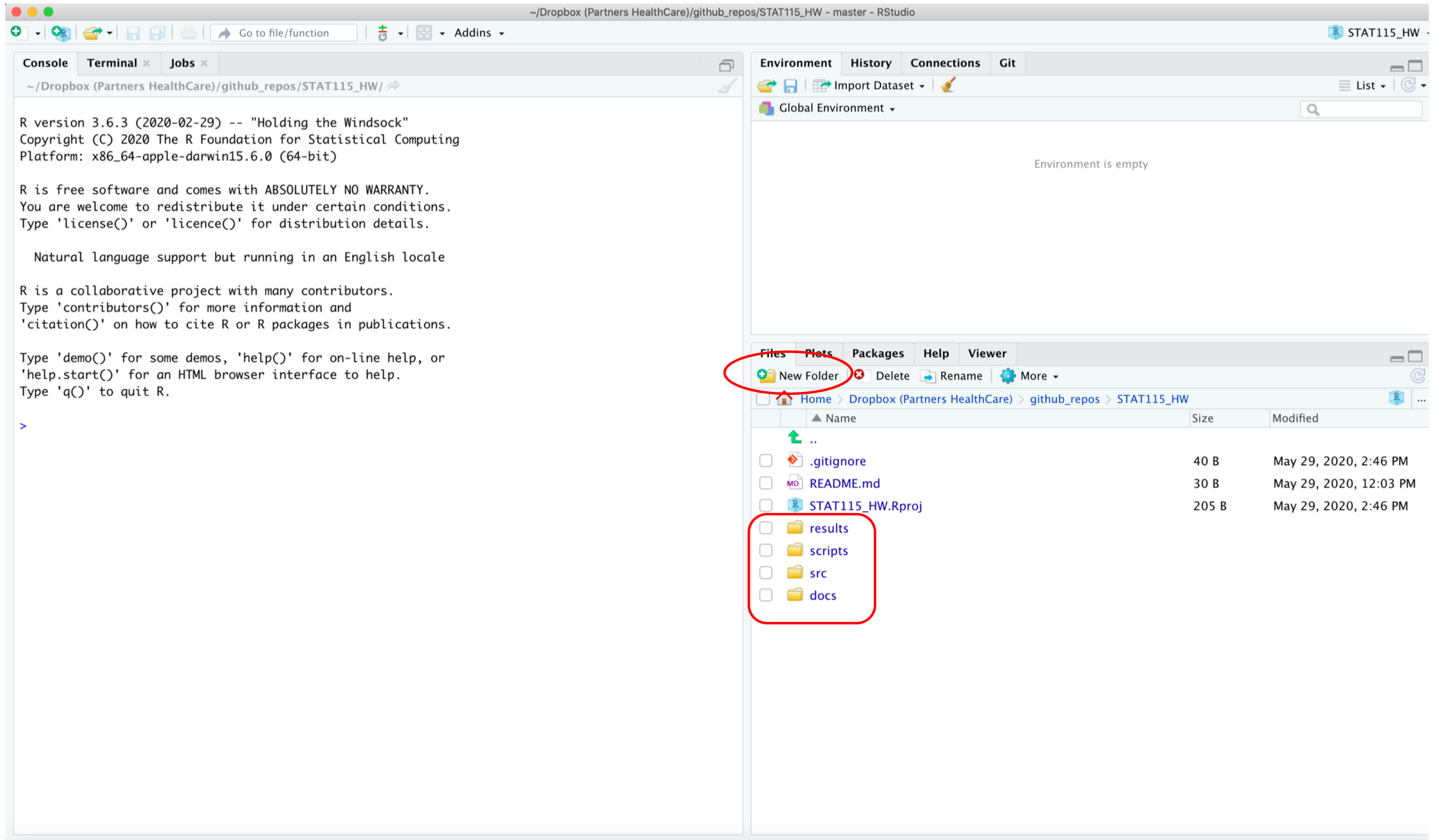
New Project

Create Project from Existing Directory

Project working directory:

~/Dropbox (Partners HealthCare)/github_repos/STAT11

In the Files tab, click New Folder and create data, results, scripts, src and docs folder

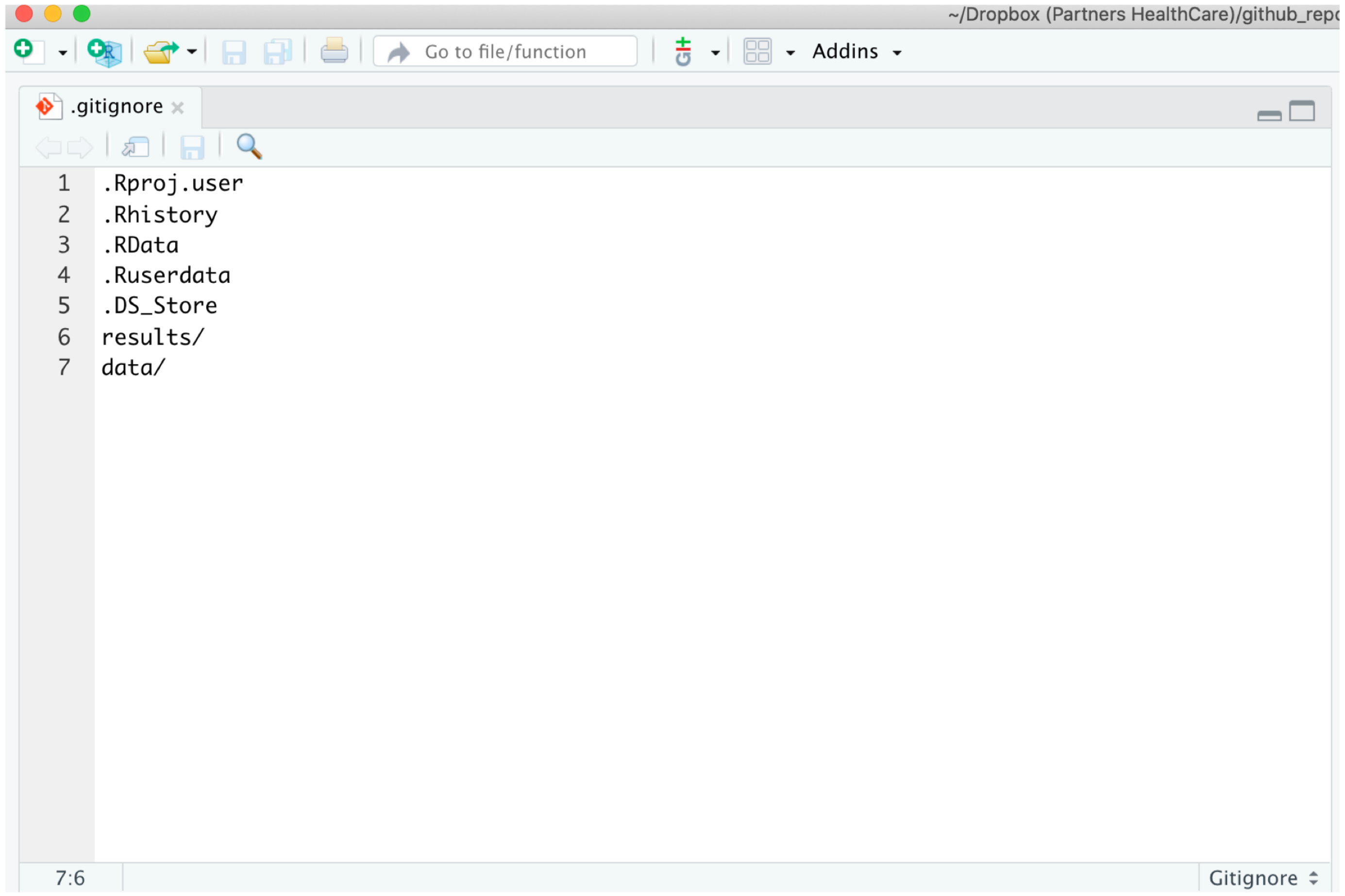


```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Name	Size	Modified
..		
.gitignore	40 B	May 29, 2020, 2:46 PM
README.md	30 B	May 29, 2020, 12:03 PM
STAT115_HW.Rproj	205 B	May 29, 2020, 2:46 PM
results		
scripts		
src		
docs		

The results folder will contain all the results obtained from the script in the scripts folder.
src folder contains R function that you can source from the script in the scripts folder.
Docs folder contains any documentations/manuscripts.

Edit the .gitignore file by clicking it



```
.Rproj.user
.Rhistory
.RData
.Ruserdata
.DS_Store
results/
data/
```

Ignore
.DS_Store file on mac

I also ask git not to track
Files in the results/ and
data/ folder since they usually
contain big files and intermediate
Files.

This how I do it, you do not have to follow.

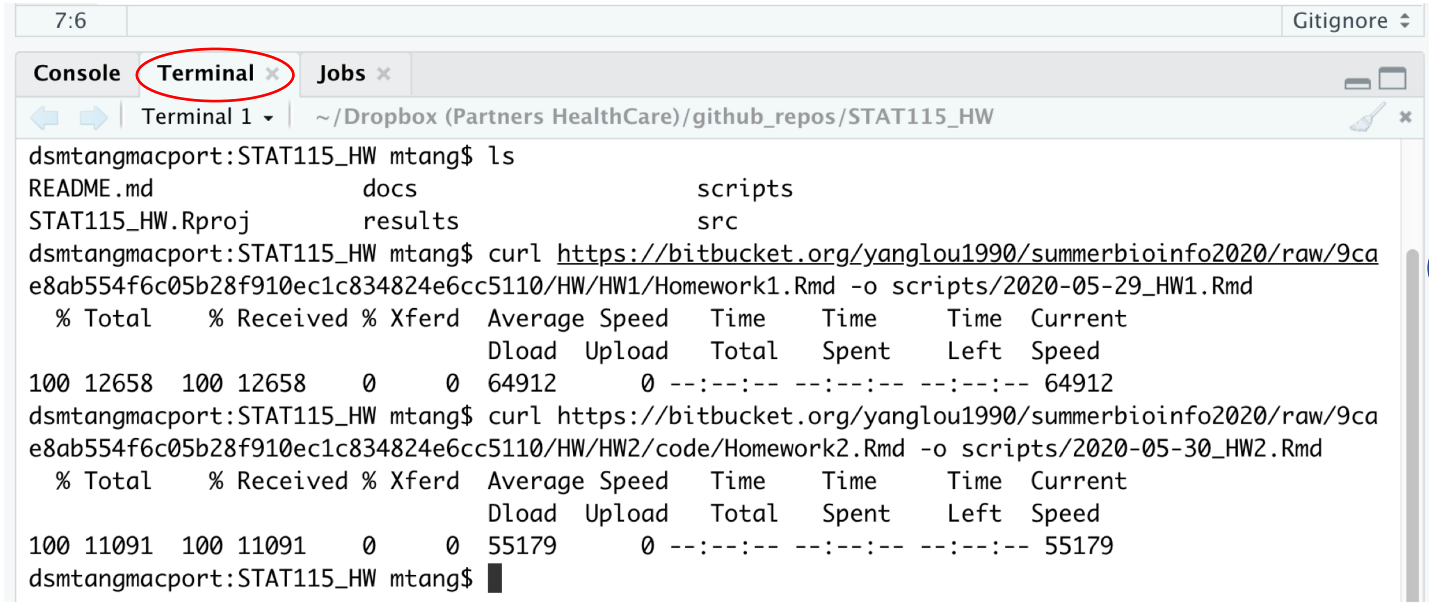
Remember I have them backed
up in dropbox if I want them.

If you want to version control
Large files, check
Git lfs https://git-lfs.github.com/

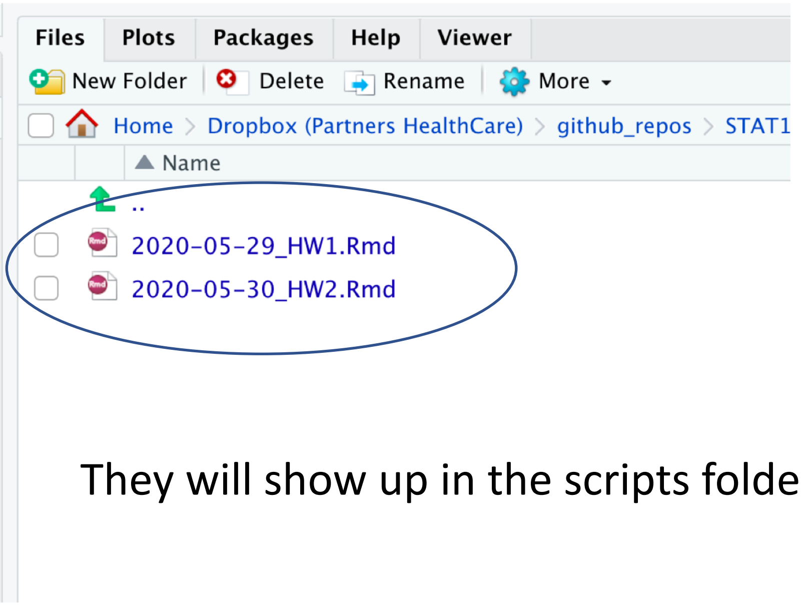
Now, you can either go to
File -- > New File -- > Rmarkdown

or download the homework Rmd file to the scripts folder.
Click Terminal tab, and use curl to download the Rmd file

Note, I renamed them by prefixing date so they are nicely sorted.



```
dsmtangmacport:STAT115_HW mtang$ ls
README.md               docs                    scripts
STAT115_HW.Rproj        results                 src
dsmtangmacport:STAT115_HW mtang$ curl https://bitbucket.org/yanglou1990/summerbioinfo2020/raw/9ca
e8ab554f6c05b28f910ec1c834824e6cc5110/HW/HW1/Homework1.Rmd -o scripts/2020-05-29_HW1.Rmd
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 12658  100 12658    0     0  64912      0 --:--:-- --:--:-- --:--:-- 64912
dsmtangmacport:STAT115_HW mtang$ curl https://bitbucket.org/yanglou1990/summerbioinfo2020/raw/9ca
e8ab554f6c05b28f910ec1c834824e6cc5110/HW/HW2/code/Homework2.Rmd -o scripts/2020-05-30_HW2.Rmd
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11091  100 11091    0     0  55179      0 --:--:-- --:--:-- --:--:-- 55179
dsmtangmacport:STAT115_HW mtang$
```

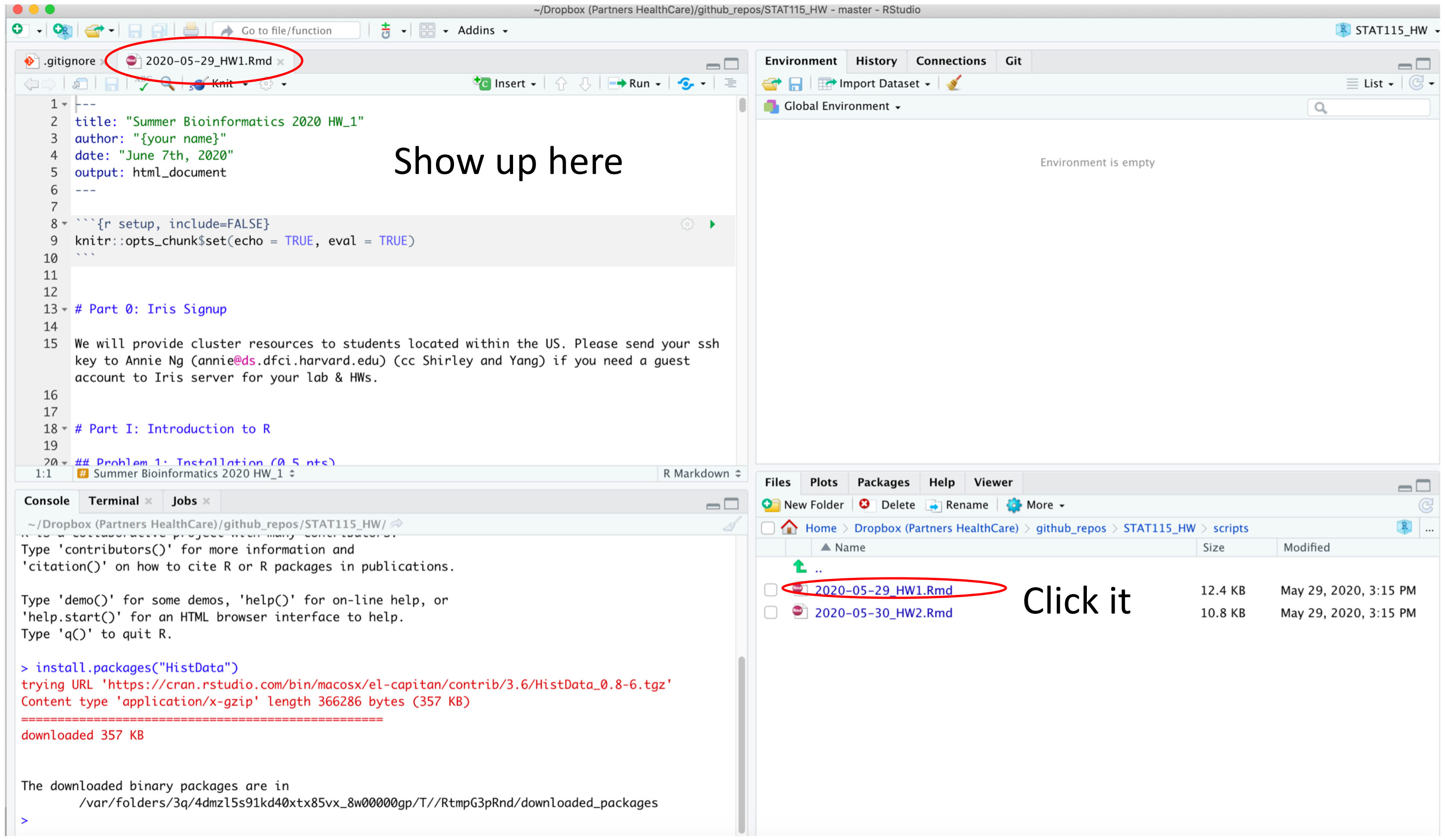


They will show up in the scripts folder

If you name Them: HW1.Rmd HW2.Rmd HW3.Rmd.

These are sorted as well, but I personally like to add date so I have an idea when did I wrote the script.
Or better to use 0 to pad the file name if you have more than 10 files so they are sorted nicely.
01_HW.Rmd
02_HW.Rmd … 10_HW.Rmd

Now, click 2020-05-29_HW1.Rmd and start to work on it.



~~~
---
title: "Summer Bioinformatics 2020 HW_1"
author: "{your name}"
date: "June 7th, 2020"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, eval = TRUE)
```


# Part 0: Iris Signup

We will provide cluster resources to students located within the US. Please send your ssh
key to Annie Ng (annie@ds.dfci.harvard.edu) (cc Shirley and Yang) if you need a guest
account to Iris server for your lab & HWs.


# Part I: Introduction to R

## Problem 1: Installation (0.5 pts)
~~~

```
~/Dropbox (Partners HealthCare)/github_repos/STAT115_HW/
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("HistData")
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/HistData_0.8-6.tgz'
Content type 'application/x-gzip' length 366286 bytes (357 KB)
==================================================
downloaded 357 KB


The downloaded binary packages are in
	/var/folders/3q/4dmzl5s91kd40xtx85vx_8w00000gp/T//RtmpG3pRnd/downloaded_packages
>
```

Name	Size	Modified
..		
2020-05-29_HW1.Rmd	12.4 KB	May 29, 2020, 3:15 PM
2020-05-30_HW2.Rmd	10.8 KB	May 29, 2020, 3:15 PM

Git version control

After you worked on the Rmd file and knitted to html, you want to push it to the github. You can either use the Rstudio built-in Git tab or use the Terminal:

- In Rstudio, click the Terminal tab:
- $ git add scripts/2020-05-29_HW1.Rmd
- $ git commit -m “homework 1 done”
- $ git push

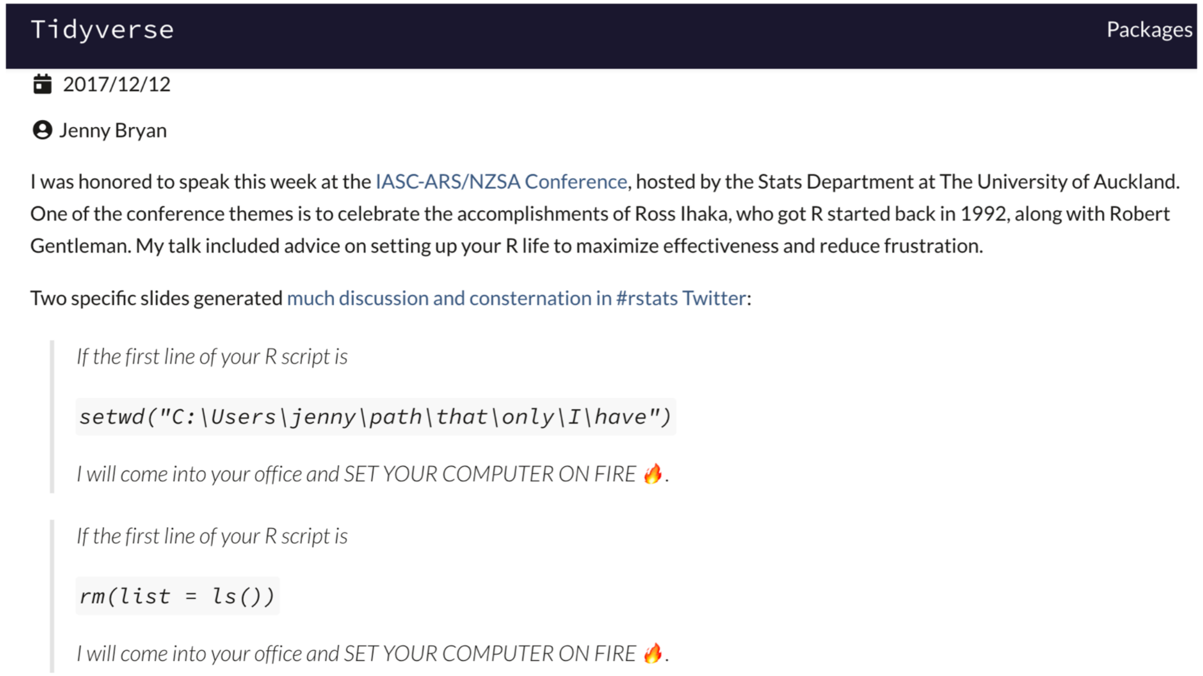
- More reading:
- Happy Git with R https://happygitwithr.com/

A different workflow

- 1. In the example, we created the github repo first → clone to local → set up Rstudio project.
- 2. if you have already created and worked on a local Rstudio project, you have to do something else:
- $ cd STAT115_HW
- $ git init
- $ git add .
- $ git commit –m “first commit”
- $ git remote add origin https://github.com/crazyhottommy/STAT115_HW.git
- $ git push -u origin master
- Reference:
- https://help.github.com/en/github/importing-your-projects-to-github/adding-an-existing-project-to-github-using-the-command-line

Other tips for working with Rstudio or R

- use here::here()
- https://www.tidyverse.org/blog/2017/12/workflow-vs-script/



Tidyverse Packages

2017/12/12

Jenny Bryan

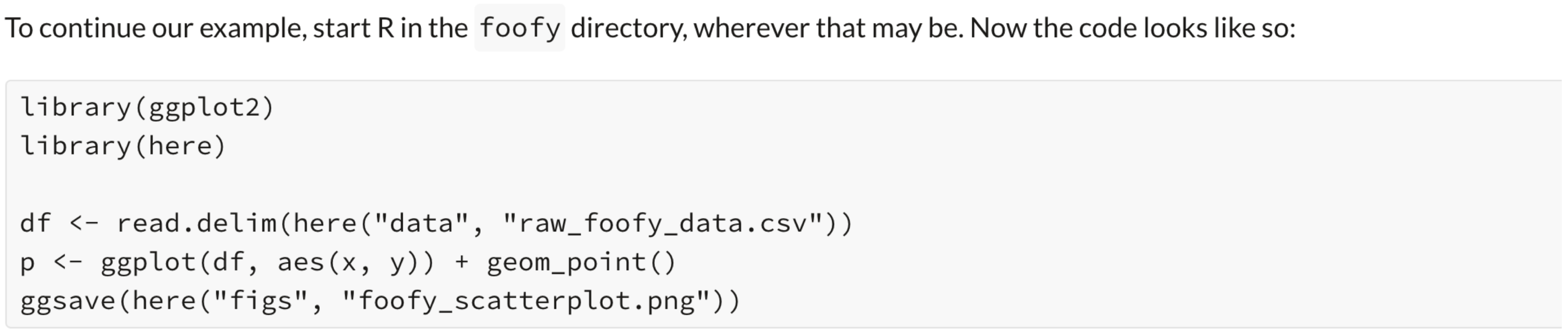
I was honored to speak this week at the IASC-ARS/NZSA Conference, hosted by the Stats Department at The University of Auckland. One of the conference themes is to celebrate the accomplishments of Ross Ihaka, who got R started back in 1992, along with Robert Gentleman. My talk included advice on setting up your R life to maximize effectiveness and reduce frustration.

Two specific slides generated much discussion and consternation in #rstats Twitter:

> *If the first line of your R script is*
>
> `setwd("C:\Users\jenny\path\that\only\I\have")`
>
> *I will come into your office and SET YOUR COMPUTER ON FIRE* 🔥.

> *If the first line of your R script is*
>
> `rm(list = ls())`
>
> *I will come into your office and SET YOUR COMPUTER ON FIRE* 🔥.

Always use here() to construct relative path. Later if you move the whole project folder into a different computer, the same code still works



To continue our example, start R in the `foofy` directory, wherever that may be. Now the code looks like so:

```
library(ggplot2)
library(here)

df <- read.delim(here("data", "raw_foofy_data.csv"))
p <- ggplot(df, aes(x, y)) + geom_point()
ggsave(here("figs", "foofy_scatterplot.png"))
```

Remember, always keep the data in the data folder untouched, I usually do

```
$ chmod u-w –R data/
```

To revoke the user’s write right so you can not edit or delete the files in the data folder.

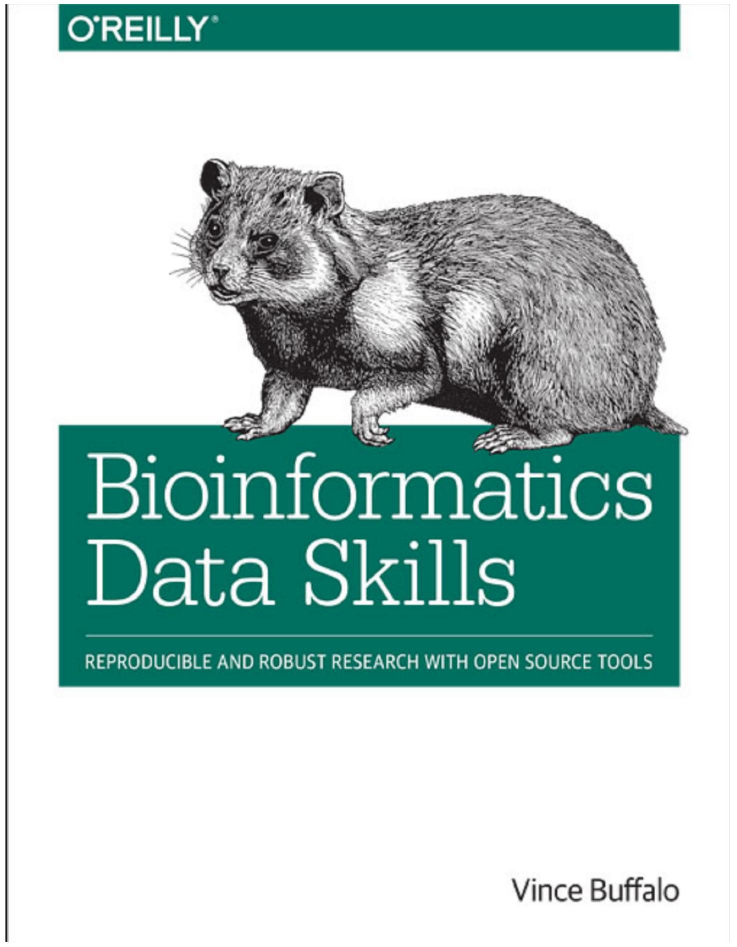
Always generate the output/intermediate files/figures in the results folder using the scripts in the scripts folder

More advanced

- A Reproducible Data Analysis Workflow with R Markdown, Git, Make, and Docker
- https://psyarxiv.com/8xzqy/

- More readings:
- **What They Forgot to Teach You About R** https://rstats.wtf/
- The renv package is a new effort to bring project-local R dependency management to your projects. https://rstudio.github.io/renv/articles/renv.html
- https://github.com/crazyhottommy/getting-started-with-genomics-tools-and-resources#automate-your-workflow-open-science-and-reproducible-research

Learn by doing, enjoy!



https://divingintogeneticsandgenomics.rbind.io/post/my-opinionated-selection-of-books-for-bioinformatics-data-science-curriculum/

What questions do you have?

Acknowledgments

Liu Lab
Shirley Liu
Yang Liu
Changxin Wan
Other lab members

Jenny Bryan
Titus Brown
Data Carpentry https://datacarpentry.org/
All the people who share their wisdom on the web
Thanks!