~~Best~~ Good Enough Practices for Reproducible Computing

Tommy Tang
Divingintogeneticsandgenomics.com
Youtube: Chatomics
@tangming2005

Why reproducibility is important?

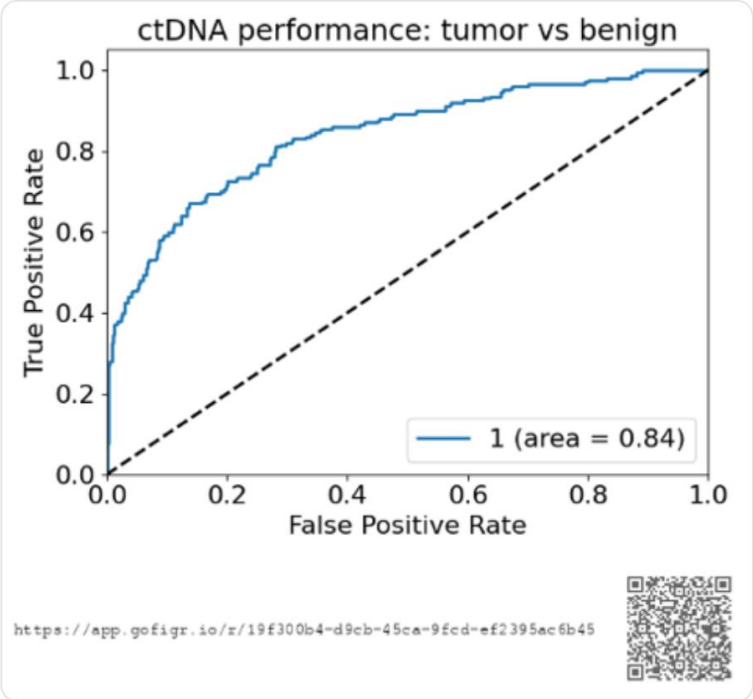
Why reproducibility is important?

Monday, February 17th

Alyssa 5:31 AM

Hi, can I have this figure broken down by cancer stage? Investor presentation at 10am

image.png



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…)

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

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.

Another example

- The Importance of Reproducible Research in High-Throughput Biology.
- 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**.

https://www.youtube.com/watch?v=7gYIs7uYbMo

What's wrong with this spreadsheet?

Supplementary Table 1. Clinical characteristics of 22 TNBC patients and somatic mutations of TNBC tumors, related to Figure 1.

Treatment	Patient ID	Biopsied lesion	Diameter of biopsied lesion (mm)	Size of target lesion / SOM (mm)	Age	Stage (TNM)	PD-L1	TILs (%)	Tumor			Blood			Relative change of target lesions (8 weeks after treatment initiation vs. Pre-	Relative change of biopsied lesions (8 weeks after treatment initiation vs. Pre-	Clinical efficacy#
									Pre-teatment	Post-treatment	Progression	Pre-teatment	Post-treatment	Progression			
Anti-PD-L1+ Chemo	P019	Lymph Node	15	36	45	rcTxN3M1	+	90	Y	Y		Y	Y		-0.67	-0.67	PR
	P010	Lung	21	35	33	rcT4NxM1	+	80	Y	N		Y	Y	Y	-0.63	-0.62	PR
	P012	Lymph Node	28	28	47	rcTxN2M0	+	70	Y	Y		Y	Y		-0.46	-0.46	PR
	P007	Lymph Node	22	22	50	rcTxNxM1	-	N	Y	N	Y	Y	Y	Y	-0.45	-0.45	PR
	P017	Lymph Node	16	45	57	cT2N2M1	-	1	Y	Y		Y	Y		-0.22	-0.22	SD
	P001	-	-	45	34	rcTxNxM1	+	30	N	N		Y	Y		0.00	-	SD
	P002	Chest Wall	48	48	59	rcT4NxM0	+	50	Y	Y		Y	Y	Y	0.00	0.00	SD
	P014	-	-	11	48	rcTxNxM1	-	N	N	N		Y	Y		0.00	-	SD
	P004	Chest Wall	35	35	45	rcT4NxM0	-	N	Y	N		Y	Y	Y	0.09	0.09	SD
	P005	Liver	87	97	52	rcTxNxM1	-	10	Y	Y		Y	Y		0.09	0.01	SD
	P016	Chest Wall	24	24	32	rcT4NxM1	-	1	Y	Y		Y	Y	Y	0.17	0.17	SD
Chemo	P022	Breast	33	48	55	cT2N2M1		2	Y	Y		Y	Y		-0.85	-0.85	PR
	P011	-	-	30	58	rcTxNxM1		<1	N	N		Y	Y		-0.67	-	PR
	P020	Breast	37	55	34	cT2N2M0		20	Y	Y		Y	Y		-0.55	-0.55	PR
	P008	Lung	22	22	64	rcTxNxM1		1	N	N		Y	Y	Y	-0.32	-0.32	PR
	P013	Liver	36	152	51	rcTxNxM1		<1	Y	Y	Y	Y	Y	Y	-0.30	-0.30	PR
	P025	Breast	26	26	53	cT2N1M0		10	Y	Y		Y	Y		-0.23	-0.23	SD
	P018	Breast	48	48	44	cT2N2M0		5	Y	Y		Y	Y		-0.09	-0.09	SD
	P023	Breast	26	42	38	cT2N2M0		20	Y	Y		Y	Y		0.03	0.03	SD
	P024	Breast	72	95	50	cT2N2M0		2	N	N		Y	Y		0.36	0.36	PD
	P003	Chest Wall	11	30	38	rcT4NxM1		<1	N	Y		N	N		2.60	7.91	PD
	P028	-	26	57	62	cT2N2M0		N	N	N		Y	N		-	-	-

N,Not available; Y,Yes

`Negative values indicate tumor shrinkage; positive values indicate tumor progression;0 indicates no change in tumor size.

#PR,partial response; SD,stable disease; PD, progressive disease

$The + sign represents censored data. These patients either have not progressed and are still being treated (P019), or the treatment and follow-up were discontinued before disease progression due to COVID-19 and other personal reasons (other marked patients).

This is a must-read for data scientists and wet-lab scientists

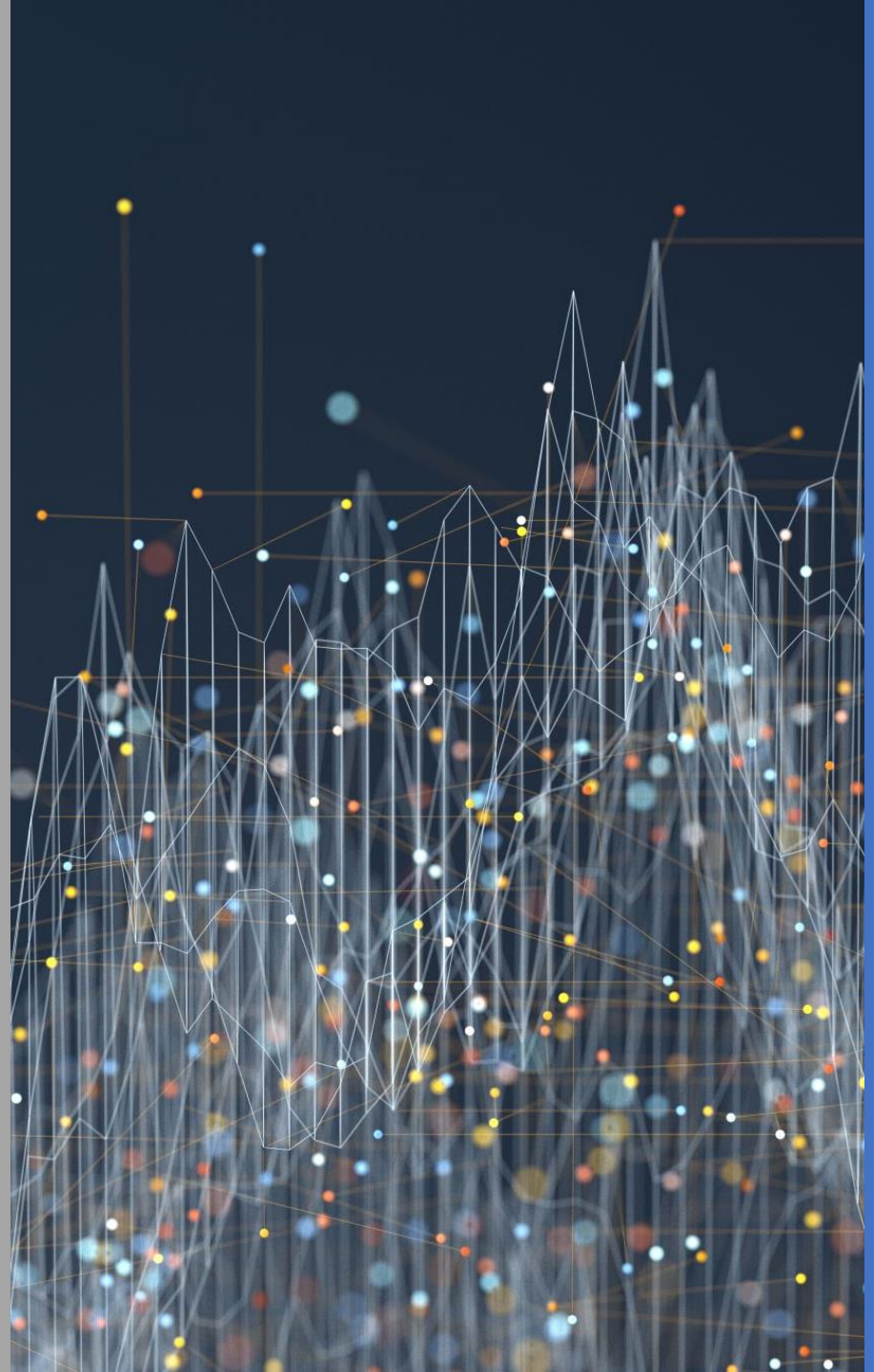
https://www.tandfonline.com/doi/full/10.1080/00031305.2017.1375989

Why reproducibility is hard?

- Raw data are not available/data are not version controlled
- Scripts are not available or available upon reasonable request ☺
- Lack of method description.
- Versions of the tools are different. (e.g. R/python/bioinformatics tools)
- Different operating systems (macOS vs Unix vs Windows).

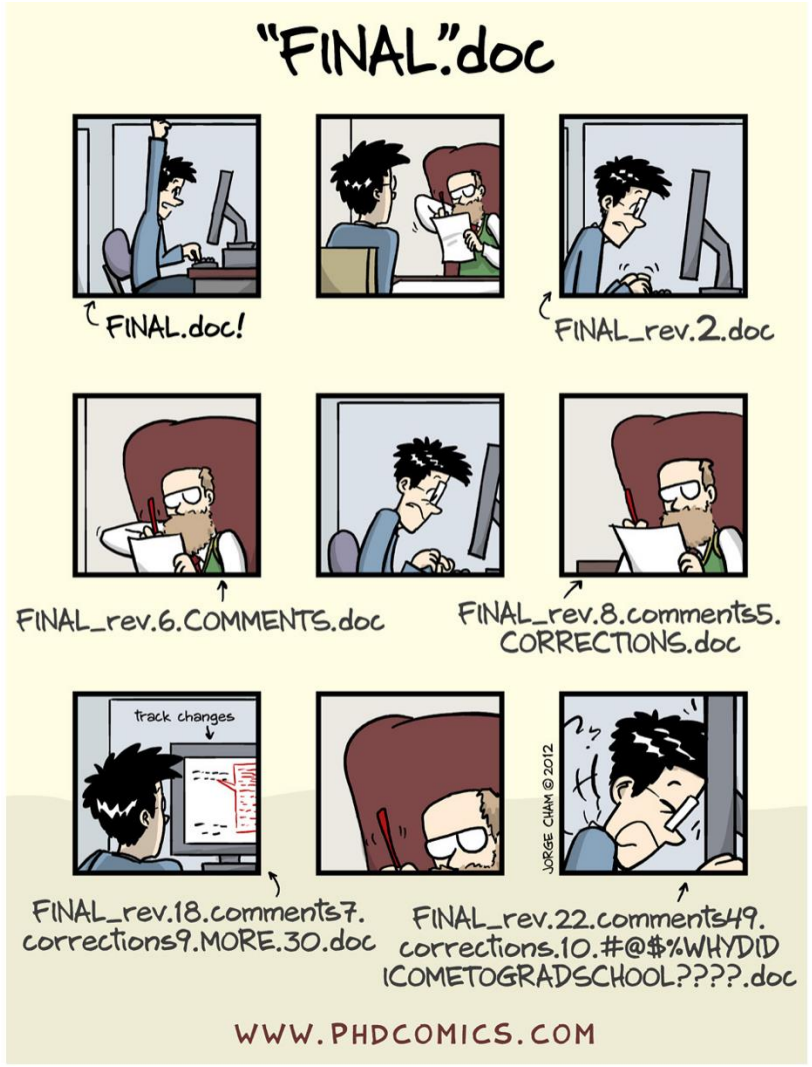
If it is so hard, should you care?

- Your closest collaborator is you six months ago
- Keep this in mind: You are going to do the same/similar analysis in the future yourself!
- Wrong decision-making for drug development can be expensive



Naming files and project organization

Naming files is hard



What are your file names look like?

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

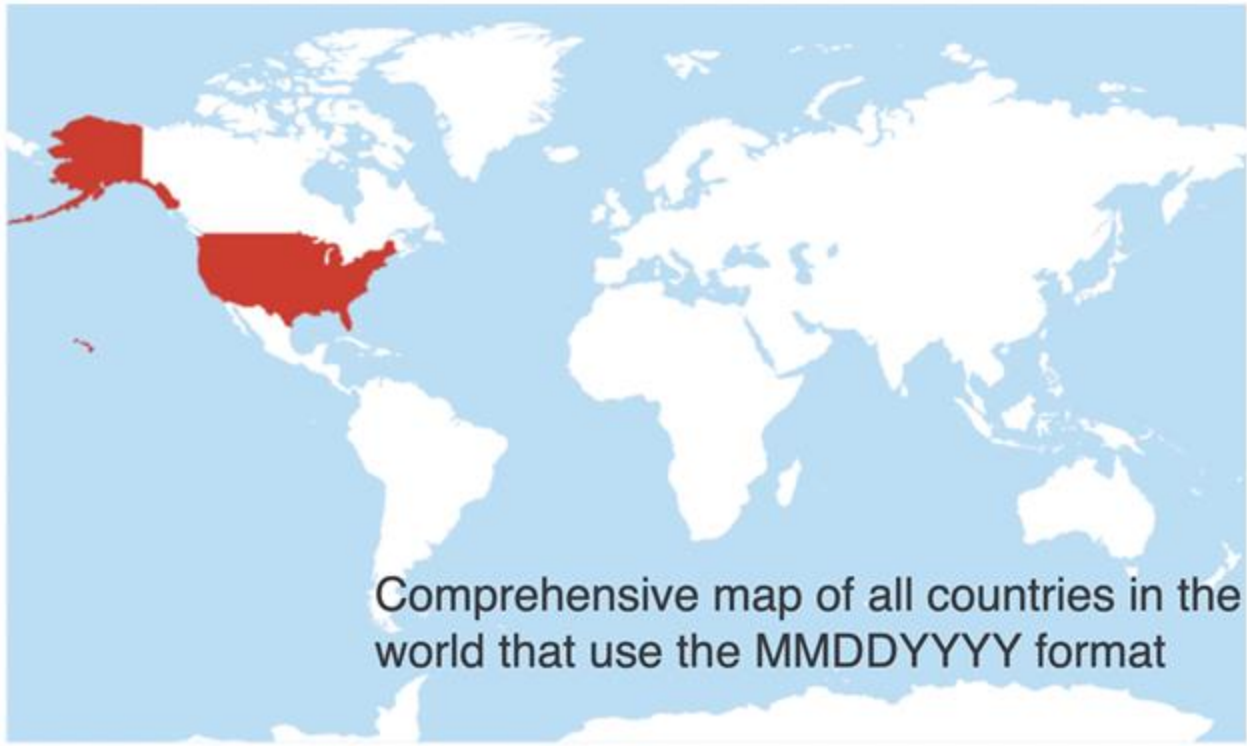
- 1. Machine readable (do not put special characters and space in the name)
- 2. Human readable (Easy to figure out what the heck something is, based on its name, add slug)
- 3. Plays well with default ordering:
    * Put something numeric first
    * Use the ISO 8601 standard for dates (YYYY-MM-DD)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

Write Dates as YYYY-MM-DD

Using the global "ISO 8601" standard, `YYYY-MM-DD`, such as `2013-02-27`.



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

```
> 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"
```

date assay sample set well

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

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

Project organization is important

PLOS | COMPUTATIONAL BIOLOGY

Browse | Publish | About

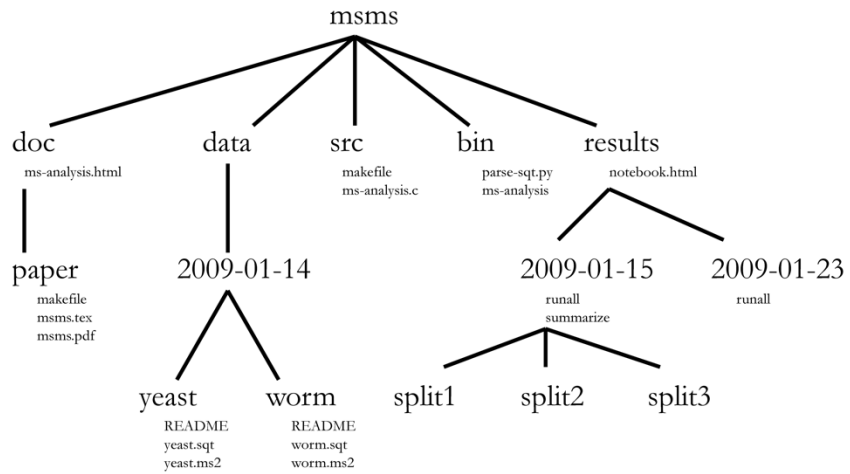
OPEN ACCESS

EDUCATION

A Quick Guide to Organizing Computational Biology Projects

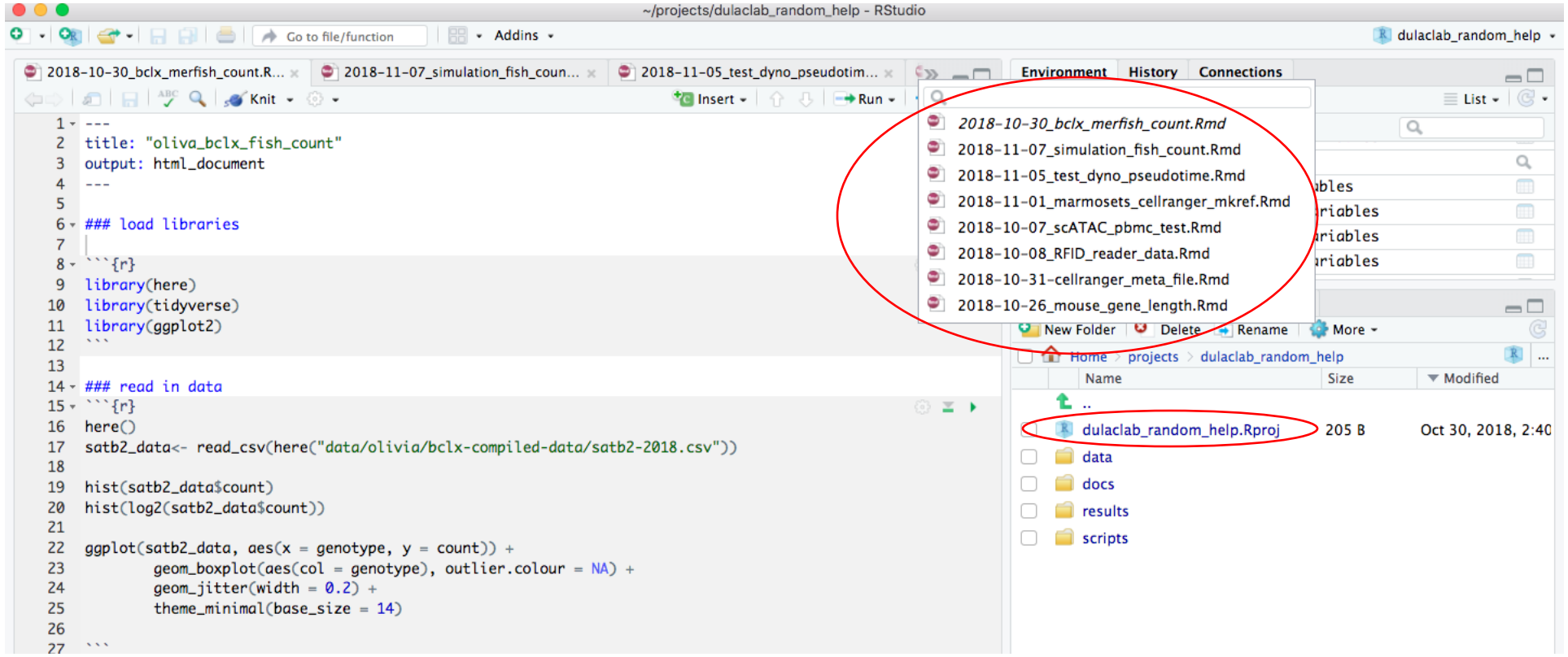
William Stafford Noble

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



You do not have to follow exactly this as long as you have a consistent folder structure for all your projects

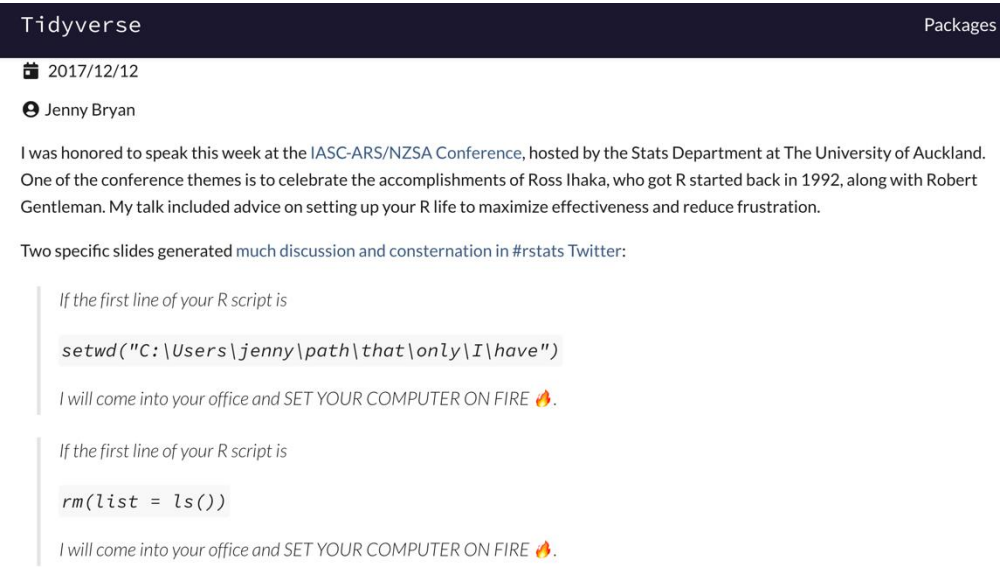
Use R project in Rstudio with consistent folder



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

Use relative path or better use here() to construct file path



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 🔥.*

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"))
```

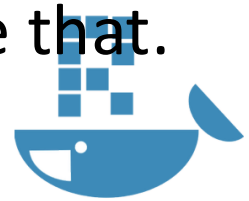
https://www.tidyverse.org/blog/2017/12/workflow-vs-script/

Use pyhere in Python

https://github.com/wildland-creative/pyhere

Docker/singularity Container

- 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.

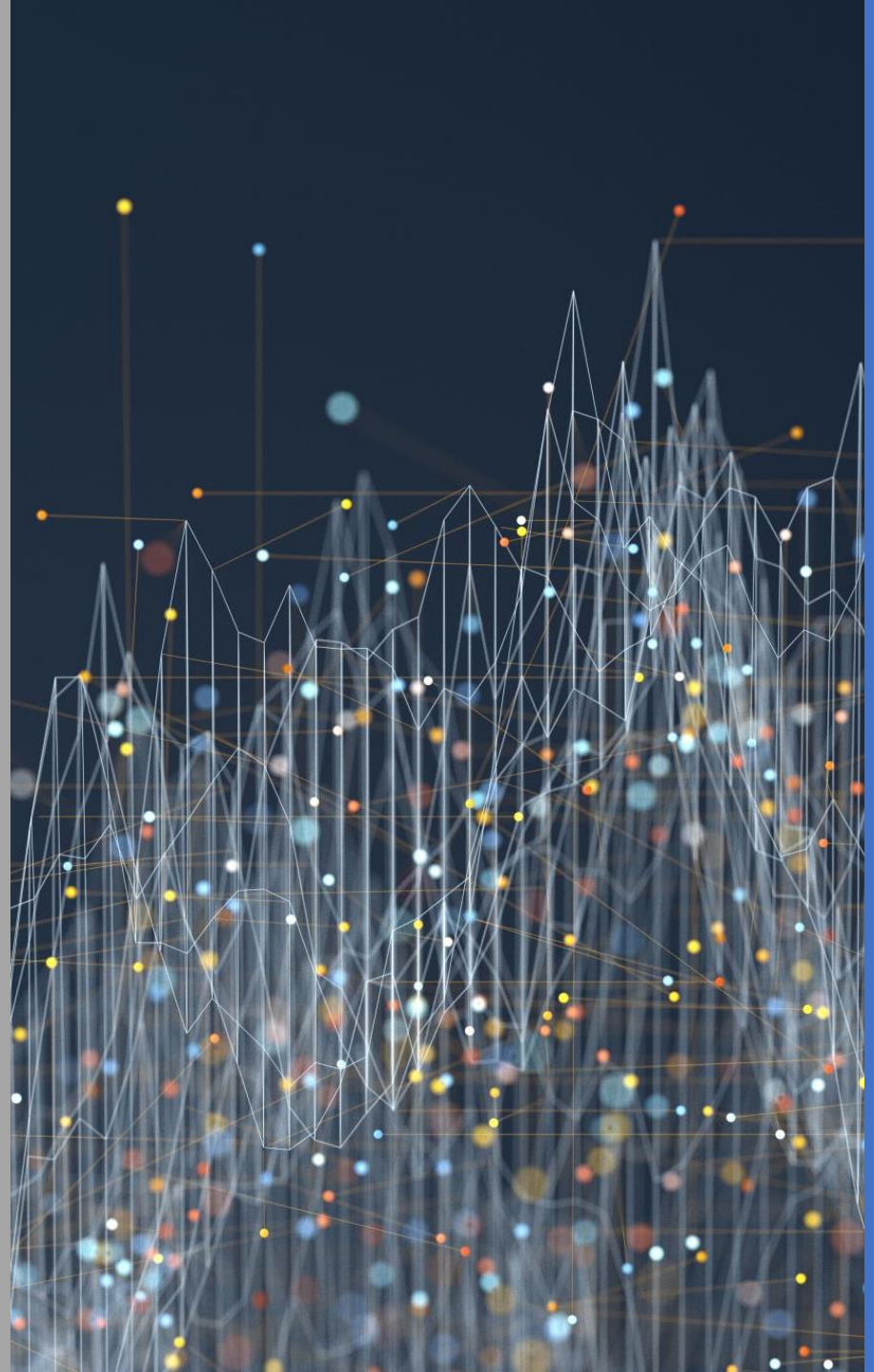


The Rocker Project

Docker Containers for the R Environment

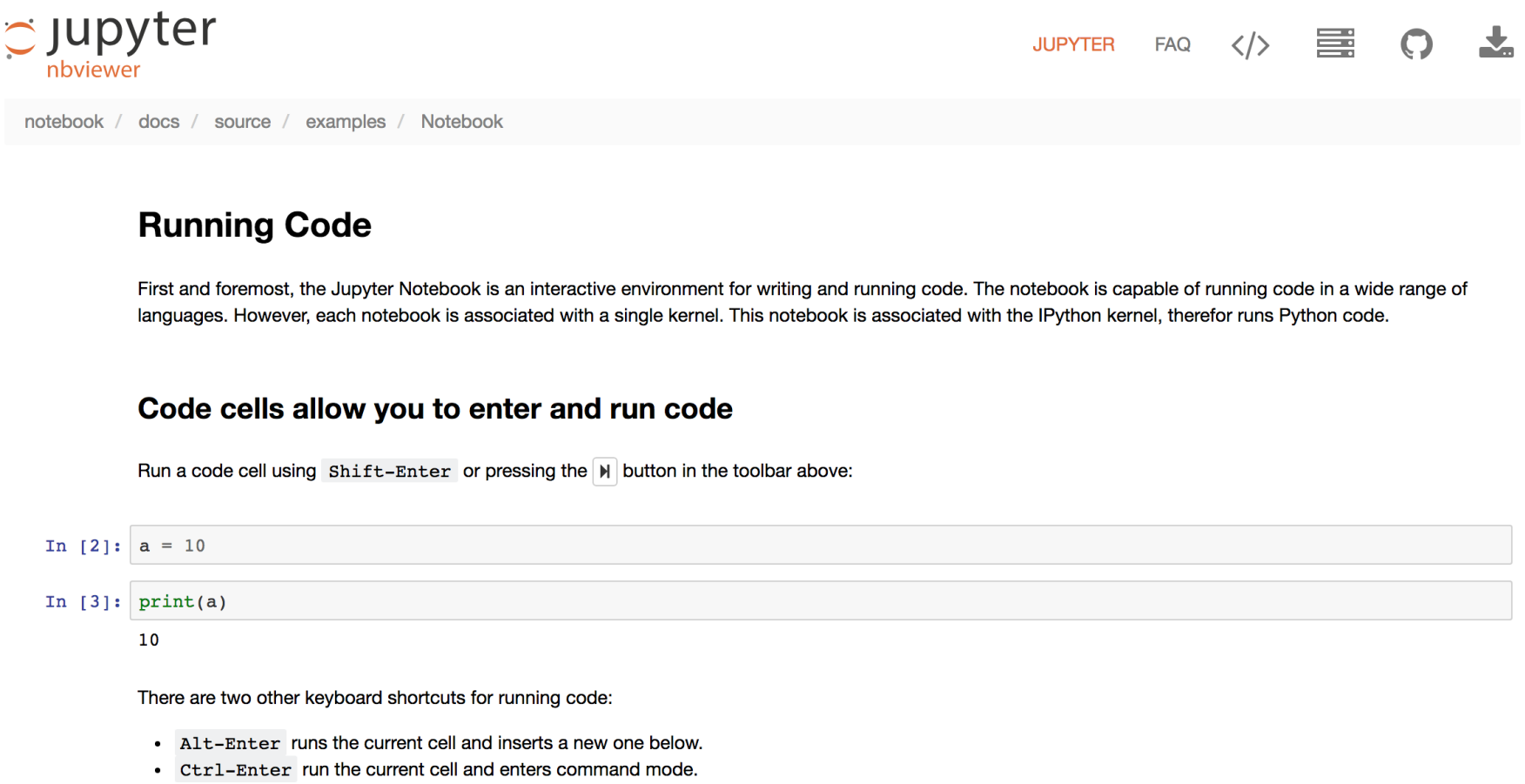
https://rocker-project.org

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



Literate programming and automation

Literate programming: mix code with prose Jupyter Notebook

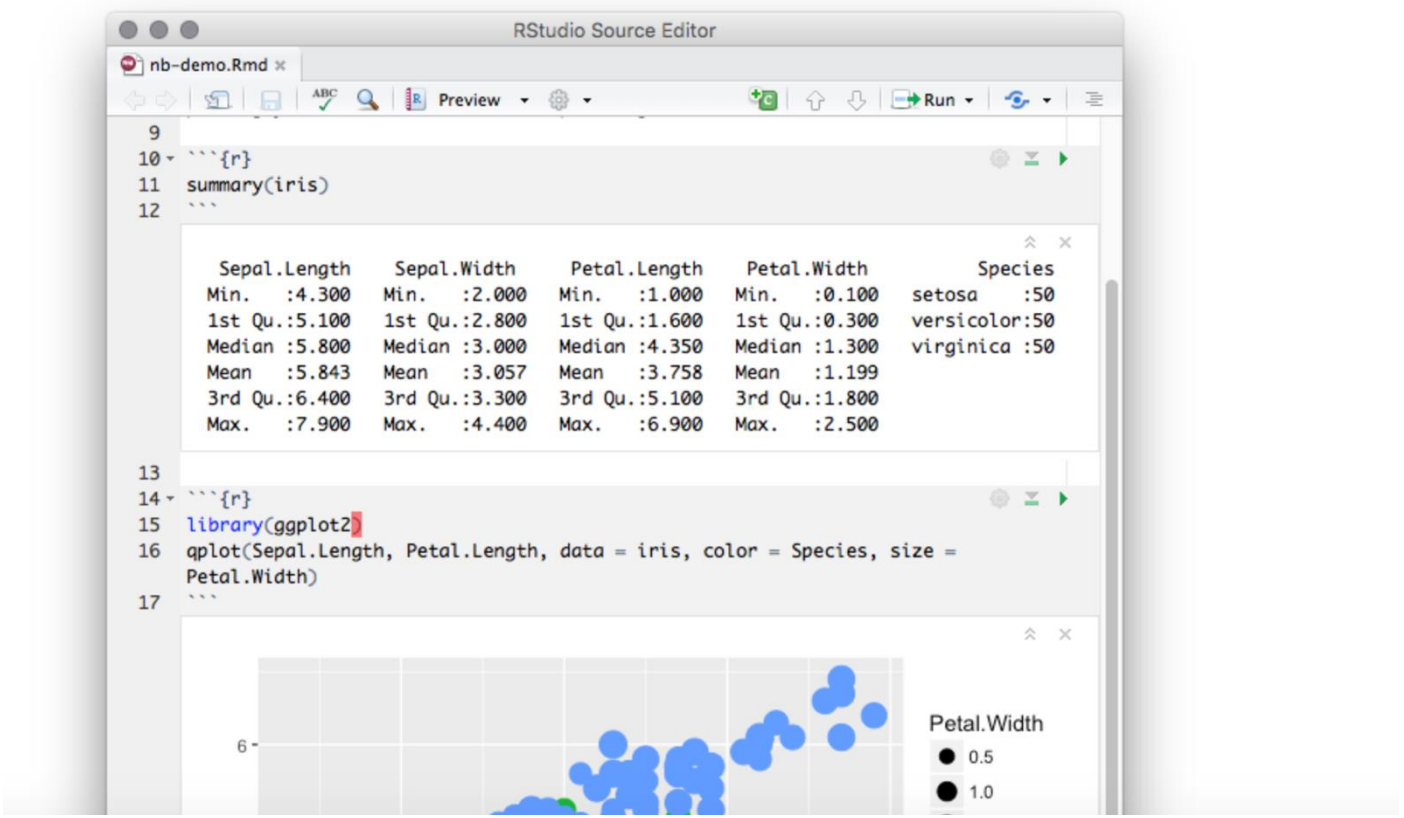


Jupyter notebook is not git friendly

R notebook/markdown

You can run python chunk
with Reticulate
https://rstudio.github.io/reticulate/

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



Quarto for literate programming

- Quarto: the next generation of Rmarkdown

Using R

Overview

Quarto is a multi-language, next generation version of R Markdown from RStudio, with many new features and capabilities. Like R Markdown, Quarto uses Knitr to execute R code, and is therefore able to render most existing Rmd files without modification.

We'll start by covering the basics of Quarto, then delve into the differences between Quarto and R Markdown in the sections on Chunk Options and Output Formats below.

Code Blocks

Code blocks that use braces around the language name (e.g. ```` ```{r} ````) are executable, and will be run by Quarto during render. Here is a simple example:

````
---
title: "ggplot2 demo"
author: "Norah Jones"
date: "5/22/2021"
format:
  html:
    code-fold: true
---

## Air Quality

@fig-airquality further explores the impact of temperature on ozone level.

```{r}
#| label: fig-airquality
#| fig-cap: "Temperature and ozone level."
#| warning: false

library(ggplot2)

ggplot(airquality, aes(Temp, Ozone)) +
  geom_point() +
  geom_smooth(method = "loess")
```
````

You'll note that there are some special comments at the top of the code block. These are cell level options that make the figure cross-referenceable.

.Rmd → .qmd

https://quarto.org

Documentation outside of the code

Tommy Tang / **Enhancer_promoter_interaction_data**

README.md

How to use the data

The data were downloaded from this paper:
Reconstruction of enhancer–target networks in 935 samples of human primary cells, tissues and cell lines

Download from http://yiplab.cse.cuhk.edu.hk/jeme/

See how I processed the data here:

1. Inside the `bed` folder: those are bed12 files you can upload to IGV or UCSC to visualize the interaction.
2. Inside the `bedpe` folder: those are bedpe files afer merging 127 ENCODE data and 800 FANTOM5 data.
3. If you need to assign your own enhancer data with a promoter, use the `ENCODE_FANTOM5_EP_refseq_promoter.tsv` inside the `annotation` folder and follow the instruction below.

asign your own enhancer data to the refseq promoter

Now, you have your own H3K27ac ChIP-seq as potential enhancers, first exclude peaks around TSS (2kb around). e.g. `my_H3K27ac_exclude_promoter.bed` . (it it should be a 4-column file: chr, start, end, info). The last column should contain some information e.g. cluster id or dummy names.

you can now cut out the first 3 columns of the `ENCODE_FANTOM5_EP_refseq_promoter.tsv` file.

```
cut -f 1-3 ENCODE_FANTOM5_EP_refseq_promoter.tsv > potential_enhancer.bed

## check how many of your enhancers have overlapping
bedtools intersect -a my_H3K27ac_exclude_promoter.bed -b potential_enhancer.bed -wa | sort | uni

#
bedtools intersect -a potential_enhancer.bed -b my_H3K27ac_exclude_promoter.bed -wa -wb > overla
```

then, use R to do a left-join of the `overlapping.tsv` with `ENCODE_FANTOM5_EP_refseq_promoter.tsv` file to get the

first commit
Tommy Tang authored Dec 21, 2017

3435fabe | History

README.md 15.98 KiB | Code | Preview

My note on how I get those files

Enhancer promoter interaction public data sets

Dnasel-seq based:

The accessible chromatin landscape of the human genome

> Genomic coordinates of all promoter DHSs and distal, non-promoter DHSs within ±500 kb correlated with them at threshold 0.7. Due to the size of this file, we are making it available through the EBI ftp server at
> ftp://ftp.ebi.ac.uk/pub/databases/ensembl/encode/integration_data_jan2011/byDataType/openchrom/jan2011/dhs_gene_connectivity/genomewideCorrs_above0.7_promoterPlusMinus500kb_withGeneNames_32celltypeCategories.bed8.gz

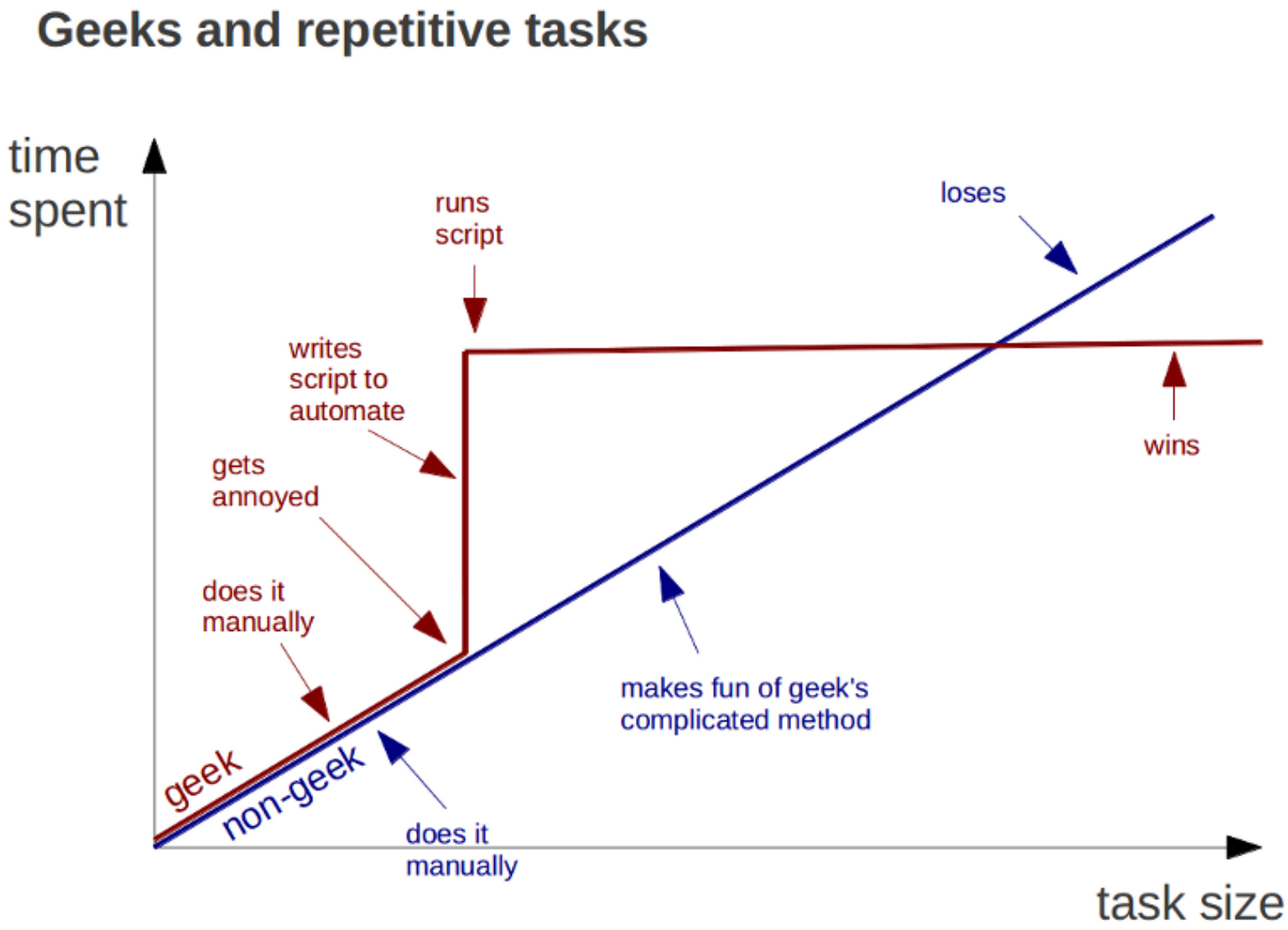
> This compressed, tab-delimited text file contains 1,672,958 lines of data, for 63,318 distinct promoter DHSs that each have at least one distal DHS connected to it. Each promoter DHS overlaps a TSS, or is the nearest DHS to the TSS in the 5′ direction; columns 1-3 contain each promoter DHS's genomic coordinates (hg19). The Gencode gene names are given in column 4. Because distinct gene names can be given to the same TSS, and because distinct TSSs can have the same nearby DHS called as their promoter DHS, data for each promoter DHS is repeated in this file roughly three times on average, with a different gene name for each repetition (there are 207,878 distinct combinations of promoter DHS + gene name in this file). Columns 5-7 contain the genomic coordinates for each diqin 500kb of the promoter DHS given in columns 1-3 that achieves correlation ≥0.7 with it; the correlation between the promoter/distal DHS pair is given in column 8. Distal DHSs appear multiple times in the file when they achieve correlation ≥0.7 with multiple promoter DHSs. Using program sort-bed from the BEDOPS genomic data analysis software suite, from the command line within a Unix system, the set of 578,905 distal DHSs connected with at least one promoter DHS can be extracted into a file named "outfile" by executing the command

```
cut -f5-7 infile | sort-bed - | uniq > outfile
```

where "infile" represents the file `genomewideCorrs_above0.7_promoterPlusMinus500kb_withGeneNames_32celltypeCategories.bed8`

https://gitlab.com/tangming2005/Enhancer_promoter_interaction_data

Automation makes your research more reproducible AND saves you time in the long run



Computers are good at repetitive work

Good Side effect of automation

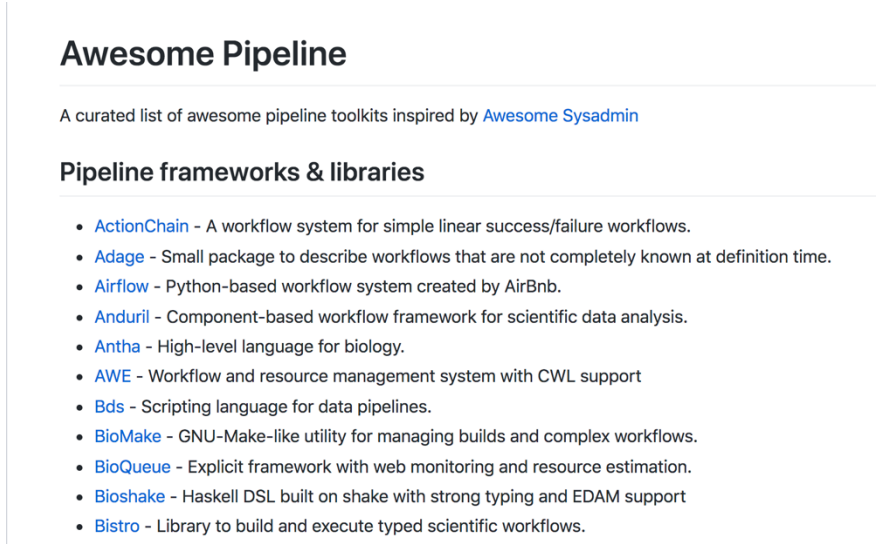
- Write scripts for everything unless it is not possible. (manual editing, document, document, document!)

- The best documentation is automation

Credit to someone in the twitter-verse ☺

Tips for automation

- If you have a repetitive simple task, put them into a shell script: my_routine.sh.
- Good old GNU make
- More recent snakemake, nextflow, WDL etc.



Awesome Pipeline

A curated list of awesome pipeline toolkits inspired by Awesome Sysadmin

Pipeline frameworks & libraries

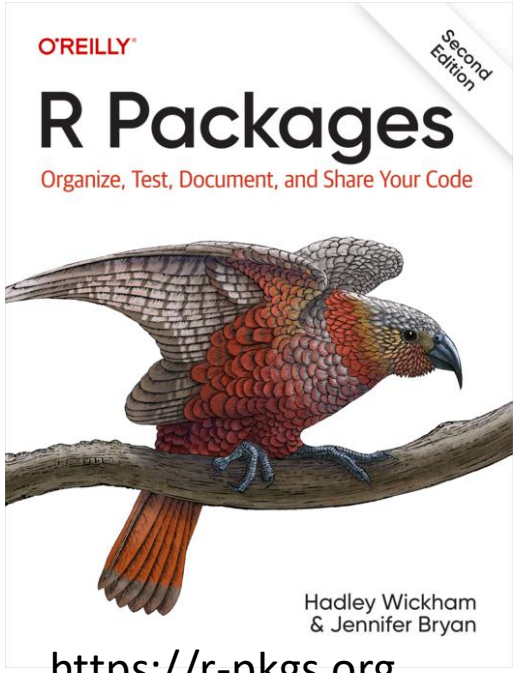
- ActionChain - A workflow system for simple linear success/failure workflows.
- Adage - Small package to describe workflows that are not completely known at definition time.
- Airflow - Python-based workflow system created by AirBnb.
- Anduril - Component-based workflow framework for scientific data 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.

{targets}
https://books.ropensci.org/targets/

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

Create an R/python package for even greater reproducibility

- Roxygen2 in R for documenting functions
- Wrap multiple functions into an R package



https://r-pkgs.org

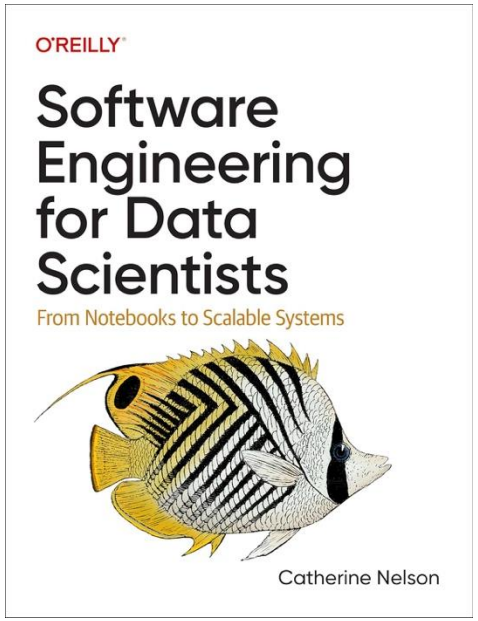
Reproducibility spectrum

- I can reproduce my own projects in my own computer in a month
- I can reproduce my own projects in 3 years
- I can reproduce my projects anywhere anytime
- Others can reproduce my projects

Good enough practices

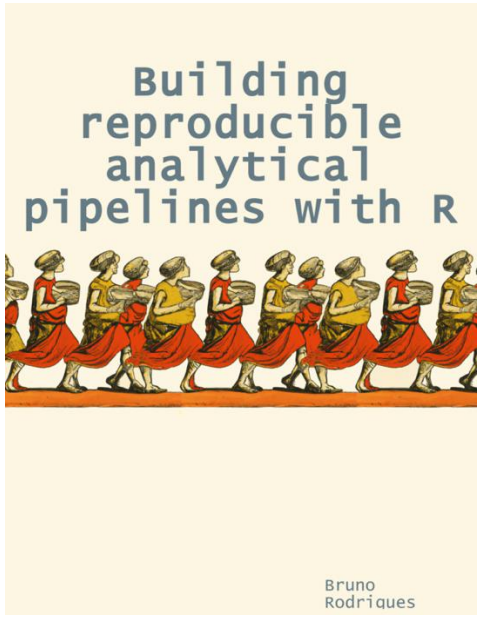
- Have a consistent folder structure for organizing Bioinformatics projects
- Automate as much as possible (e.g.,pre-processing)
- Use notebook (Rmarkdown or Juypter notebook) combining code and documentation
- Write functions to avoid repetition (use LLM)
- Version control code with git
- Use docker, conda, uv, renv to manage your computing environment
- Knitr to create HTML report
- Create a slide deck for each analysis. Put the github link at the end of the slide

Further reading

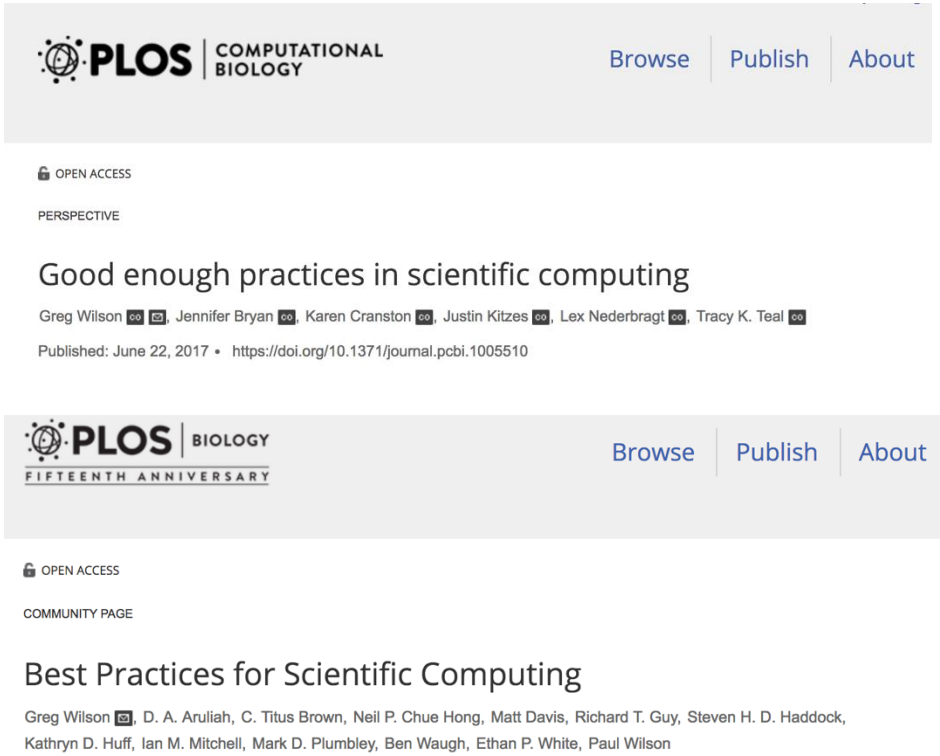


Unit test

CI/CD



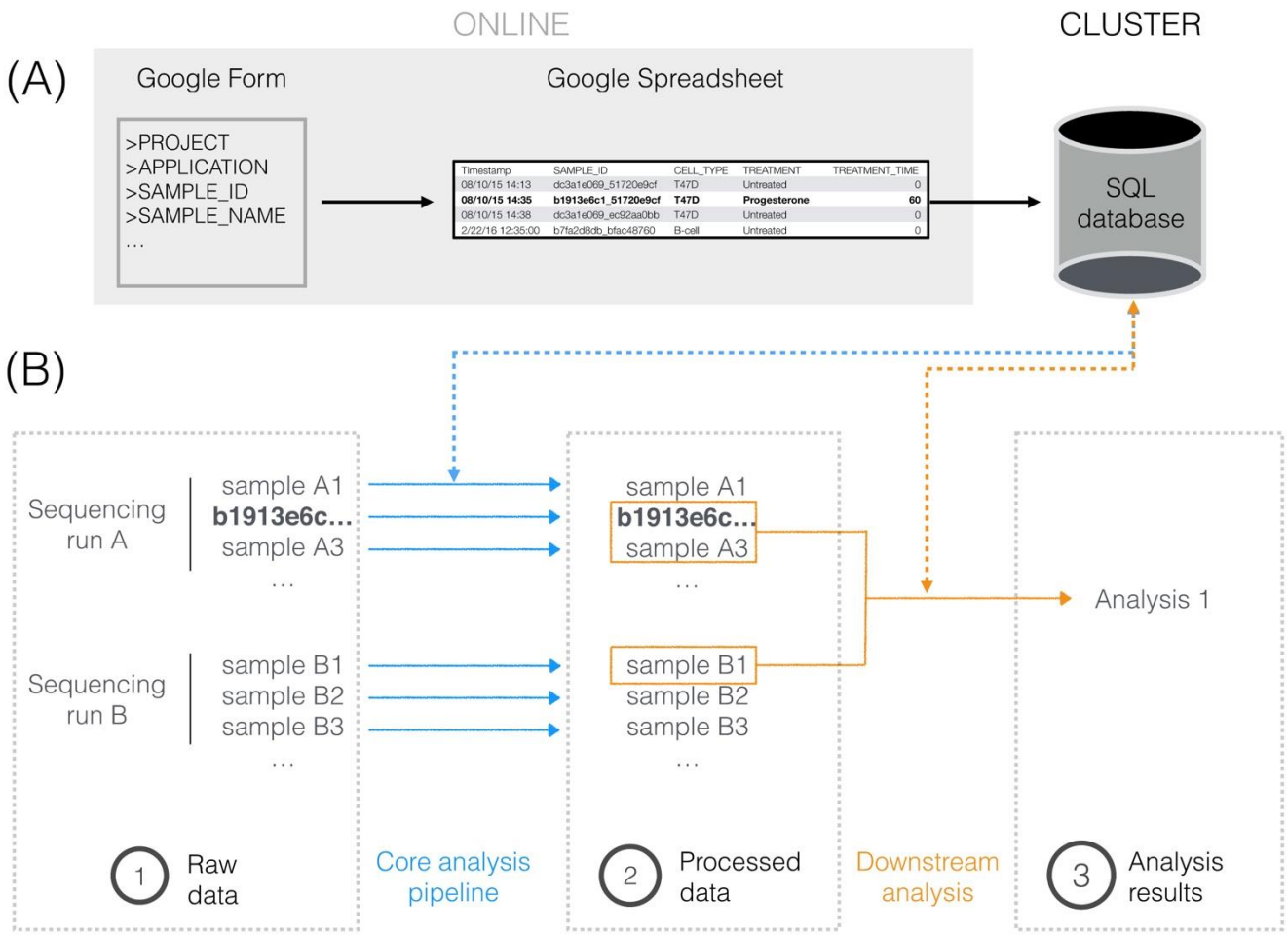
https://raps-with-r.dev



What questions do you have?

Backup

What makes large sequencing project successful



Be cautious with Excel

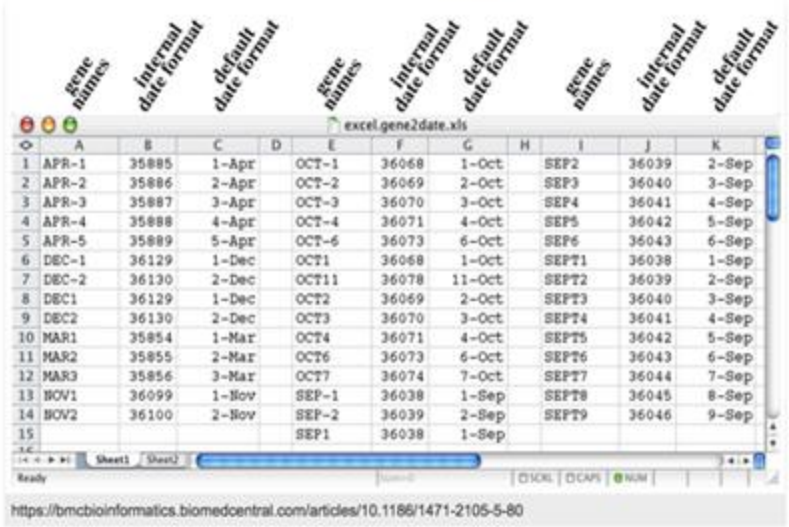
Comment | Open Access | Published: 23 August 2016

Gene name errors are widespread in the scientific literature

Mark Ziemann, Yotam Eren & Assam El-Osta

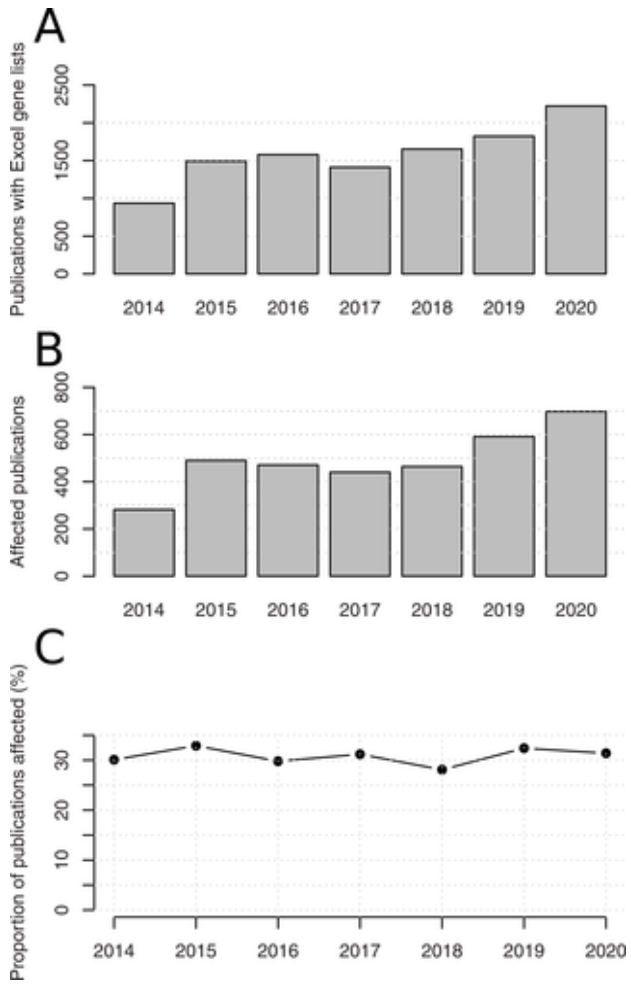
Genome Biology **17**, Article number: 177 (2016) | Cite this article

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



https://www.theverge.com/2020/8/6/21355674/human-genes-rename-microsoft-excel-misreading-dates

Gene name errors: Lessons not learned

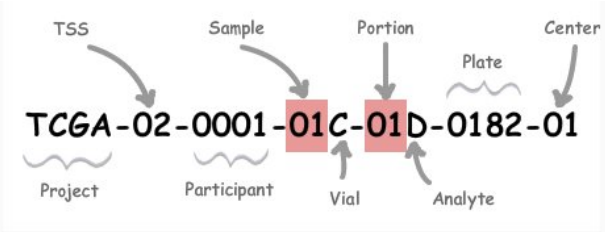


https://www.sciencedirect.com/science/article/pii/S0018506X18302599?via%3Dihub

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

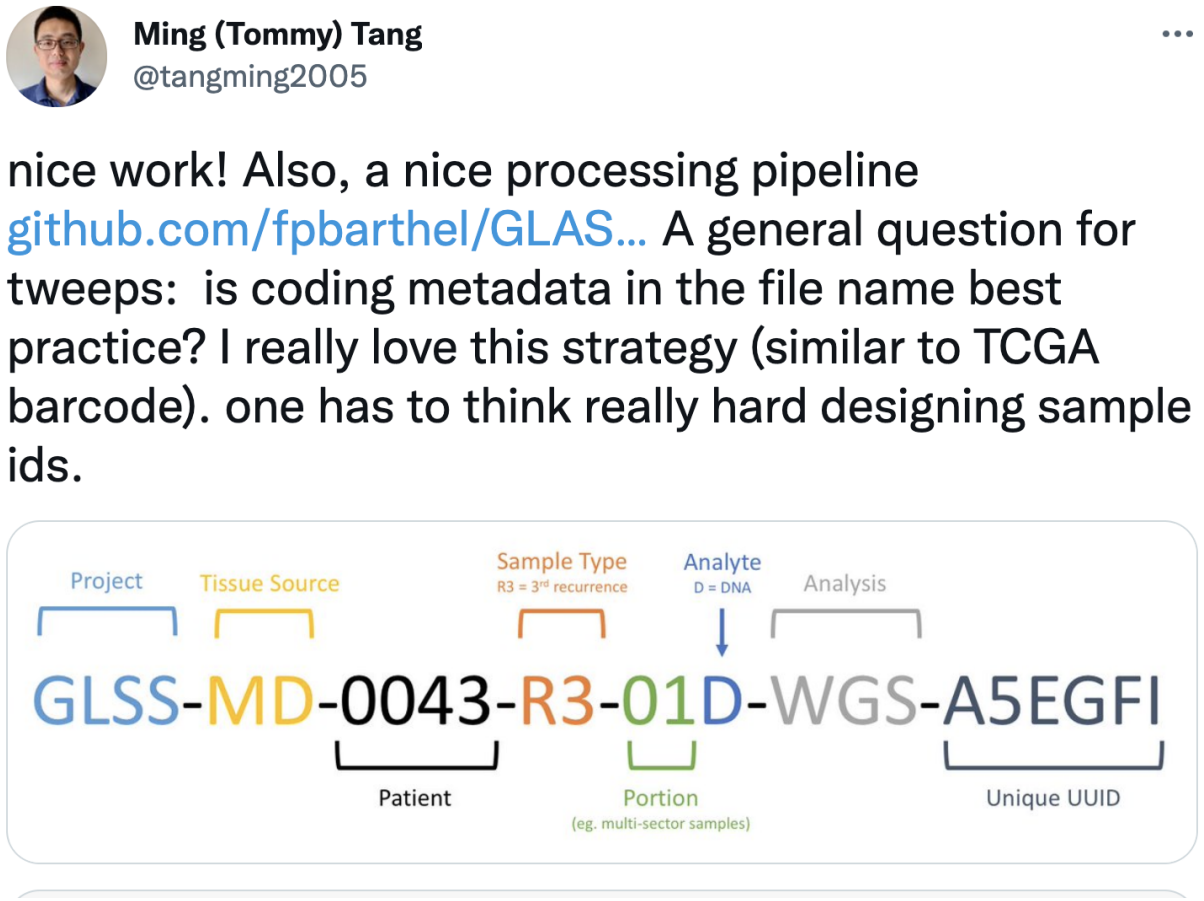
https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008984

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

Good idea to encode metadata to filenames?



Ming (Tommy) Tang
@tangming2005

nice work! Also, a nice processing pipeline github.com/fpbarthel/GLAS... A general question for tweeps: is coding metadata in the file name best practice? I really love this strategy (similar to TCGA barcode). one has to think really hard designing sample ids.

Jeremy Leipzig @jermdemo · May 27
Replying to @tangming2005
Putting metadata in a filename is bad practice in the same sense as leaving your sleeping toddler in the car while you run to the ATM. What else are you going to do?
1 2

Ming (Tommy) Tang @tangming2005 · May 27
want to hear more on why? I know it might be bad to leak private information if code the metadata in the filename. on the other hand, working with a filename of uuid.txt is not fun (I know it is designed for machine not human).
2

Jeremy Leipzig @jermdemo · May 27
If the metadata is wrong you need to change the filename and change it everywhere it might have been referenced. Also some pipeline frameworks don't respect filenames to the same extent you might.
2 1

Jeff Gentry @geoffjentry · May 27
💯- seems like a good idea until it's not and then you're hosed
1

https://twitter.com/tangming2005/status/1398094371594149893

This is a must-read for data scientists and wet-lab scientists

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

Tidy data

Structuring data in spreadsheets

The cardinal rule of using spreadsheet programs for data is to keep it "tidy":

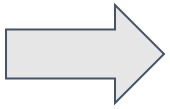
1. Put all your variables in columns - the thing you're measuring, like 'weight' or 'temperature'.
2. Put each observation in its own row.
3. Don't combine multiple pieces of information in one cell. Sometimes it just seems like one thing, but think if that's the only way you'll want to be able to use or sort that data.
4. Leave the raw data raw - don't change it!
5. Export the cleaned data to a text-based format like CSV (comma-separated values) format. This ensures that anyone can use the data, and is required by most data repositories.

Hadley Wickham, *Tidy Data*, Vol. 59, Issue 10, Sep 2014, Journal of Statistical Software. http://www.jstatsoft.org/v59/i10.

Tidy data

B

	A	B	C	D	E	F	G	H	I
1		1 min				5 min			
2	strain	normal		mutant		normal		mutant	
3	A	147	139	166	179	334	354	451	474
4	B	246	240	178	172	514	611	412	447



	A	B	C	D	E
1	strain	genotype	min	replicate	response
2	A	normal	1	1	147
3	A	normal	1	2	139
4	B	normal	1	1	246
5	B	normal	1	2	240
6	A	mutant	1	1	166
7	A	mutant	1	2	179
8	B	mutant	1	1	178
9	B	mutant	1	2	172
10	A	normal	5	1	334
11	A	normal	5	2	354
12	B	normal	5	1	514
13	B	normal	5	2	611
14	A	mutant	5	1	451
15	A	mutant	5	2	474
16	B	mutant	5	1	412
17	B	mutant	5	2	447

Make the data in a tidy format which is ggplot2 friendly!

Fill in empty values with tidyr

```
# direction = "down" -------------------------
# Value (year) is recorded only when it changes
sales <- tibble::tribble(
  ~quarter, ~year, ~sales,
  "Q1",    2000,    66013,
  "Q2",      NA,    69182,
  "Q3",      NA,    53175,
  "Q4",      NA,    21001,
  "Q1",    2001,    46036,
  "Q2",      NA,    58842,
  "Q3",      NA,    44568,
  "Q4",      NA,    50197,
  "Q1",    2002,    39113,
  "Q2",      NA,    41668,
  "Q3",      NA,    30144,
  "Q4",      NA,    52897,
  "Q1",    2004,    32129,
  "Q2",      NA,    67686,
  "Q3",      NA,    31768,
  "Q4",      NA,    49094
)
```

```
# `fill()` defaults to replacing missing data from top to bottom
sales %>% fill(year)
#> # A tibble: 16 × 3
#>    quarter  year sales
#>    <chr>   <dbl> <dbl>
#>  1 Q1       2000 66013
#>  2 Q2       2000 69182
#>  3 Q3       2000 53175
#>  4 Q4       2000 21001
#>  5 Q1       2001 46036
#>  6 Q2       2001 58842
#>  7 Q3       2001 44568
#>  8 Q4       2001 50197
#>  9 Q1       2002 39113
#> 10 Q2       2002 41668
#> 11 Q3       2002 30144
#> 12 Q4       2002 52897
#> 13 Q1       2004 32129
#> 14 Q2       2004 67686
#> 15 Q3       2004 31768
#> 16 Q4       2004 49094
```

tidyr::fill(data, ..., .direction = c("down", "up", "downup", "updown"))

Use the right Null values

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	Alternative form of NA, but often not compatible with software		Avoid
NULL	Can cause problem with data type		Avoid
None	Uncommon. Can cause problem with data type.	Python	Avoid
No data	Uncommon. Can cause problem with data type, Contains a space		Avoid
Missing	Uncommon. Can cause problem with data type.		Avoid
-,+,.	Uncommon. Can cause problem with data type.		Avoid

Use good 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.

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

read_tsv() %>% janitor::clean_names()

Be consistent

Use consistent codes for categorical variables. For a categorical variable like the sex of a mouse in a genetics study, use a single common value for males (e.g., "`male`"), and a single common value for females (e.g., "`female`"). Do not sometimes write "`M`," sometimes "`male`," and sometimes "`Male`." Pick one and stick to it

Use consistent variable names. If in one file (e.g., the first batch of subjects), you have a variable called "`Glucose_10wk`," then call it exactly that in other files (e.g., for other batches of subjects). If it is variably called "`Glucose_10wk`," "`gluc_10weeks`," and "`10 week glucose`," then downstream the data analyst will have to work out that these are all really the same thing.

Use consistent subject identifiers. If sometimes it is "`153`" and sometimes "`mouse153`" and sometimes "`mouse-153F`" and sometimes "`Mouse153`," there is going to be extra work to figure out who is who.

```
```{r}

LUAD_index<- !is.na(TCGA_LUAD_data$paper_expression_subtype)
LUSC_index<- !is.na(TCGA_LUSC_data$paper_Expression.Subtype)
```
```

Common mistakes

- https://datacarpentry.org/spreadsheet-ecology-lesson/02-common-mistakes/

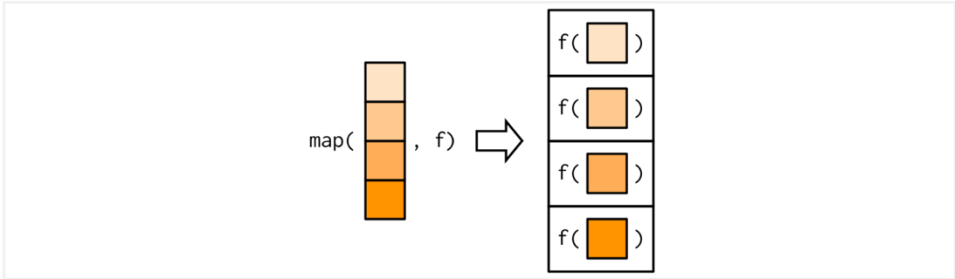
  “There are a few potential errors to be on the lookout for in your own data as well as data from collaborators or the Internet. If you are aware of the errors and the possible negative effect on downstream data analysis and result interpretation, it might motivate yourself and your project members to 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**”

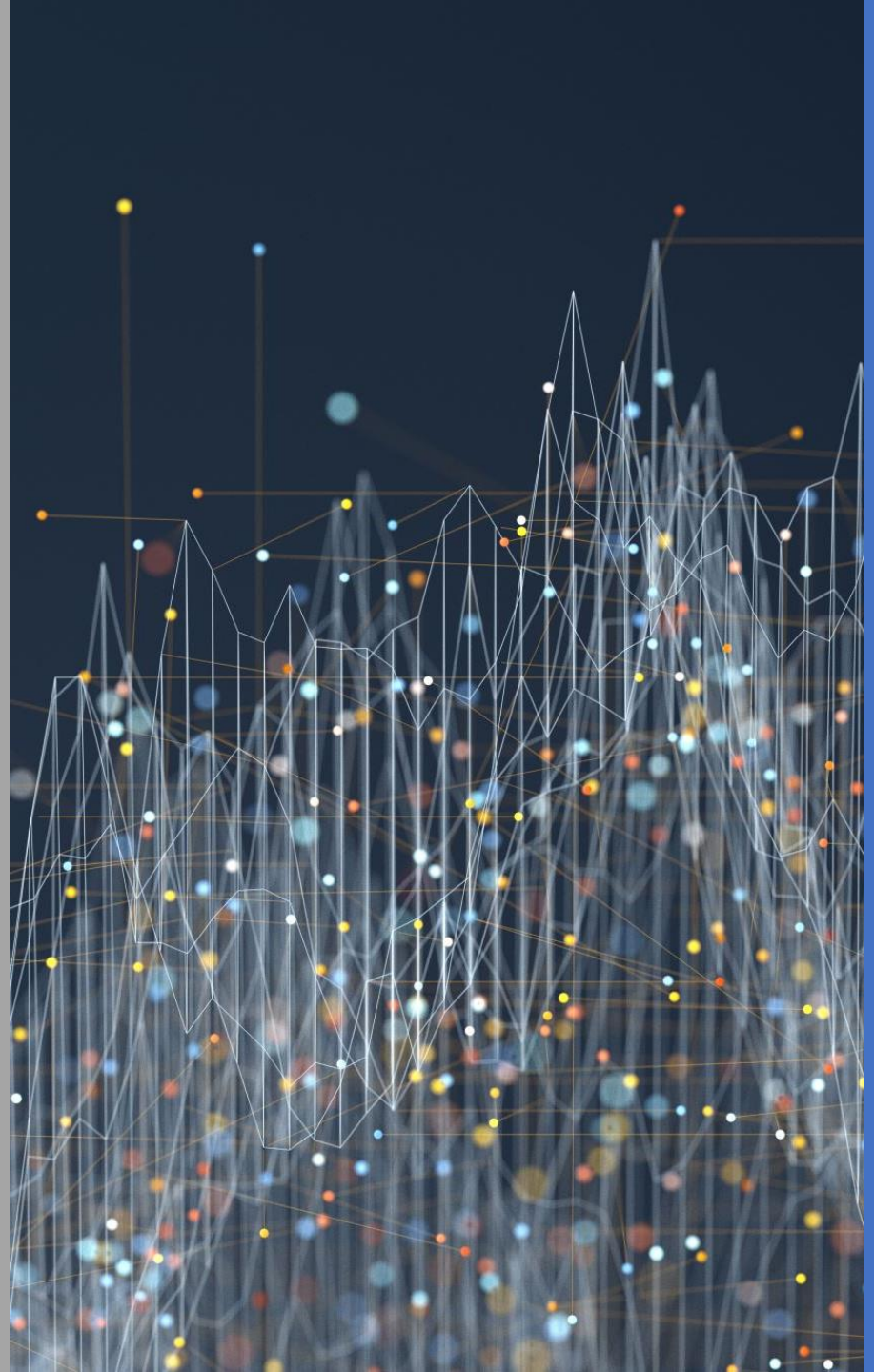
Functional programming (do not repeat yourself)

```r
ER_bw<- import(here("data/Capiva_cut_run_internal/ER/ER-DMSO-1.bw"))
ER_bw_Capi<- import(here("data/Capiva_cut_run_internal/ER/ER-Cap-1.bw"))
ER_bw_Flu<- import(here("data/Capiva_cut_run_internal/ER/ER-Fulv-1.bw"))
ER_bw_CF<- import(here("data/Capiva_cut_run_internal/ER/ER-CF-1.bw"))


ATAC_bw<- import(here("data/Capiva_cut_run_internal/ATAC/DMSO_MCF7_1.bw"))
ATAC_Capi<- import(here("data/Capiva_cut_run_internal/ATAC/Cap_MCF7_1.bw"))
ATAC_Flu<- import(here("data/Capiva_cut_run_internal/ATAC/Fulv_MCF7_1.bw"))
ATAC_bw_CF<- import(here("data/Capiva_cut_run_internal/ATAC/CF_MCF7_1.bw"))
```

```r
H3K4me1_bw_files<- list.files(here("data/H3K4me1"), full.name=TRUE, pattern= "bw$")[c(1,4,7,10)]

names(H3K4me1_bw_files)<- basename(H3K4me1_bw_files) %>%
  str_replace("([0-9A-Za-z]_K4me1)_.+", "\\1")

H3K4me1_bws<- purrr::map(H3K4me1_bw_files, import)


KMT2D_bw_files<- list.files(here("data/KMT2D"), full.name = TRUE, pattern = "bw$")[c(1,4,7,10)]

names(KMT2D_bw_files)<- basename(KMT2D_bw_files) %>%
  str_replace("([0-9A-Za-z]_KMT2D)_.+", "\\1")

KMT2D_bws<- purrr::map(KMT2D_bw_files, import)
```

I am repeating myself





Git version control and computing environment management

Why use git

- Recover code
- Collaboration or Inheriting other people’s project
- Code review (we all make mistakes)
- Use branches (that can be throw away)

Git is hard?

- Six basic commands can take you a long way:
  - Git init
  - Git clone
  - Git add
  - Git commit
  - Git push
  - Git pull

- Commit often, push by the end of the day

https://happygitwithr.com
https://learngitbranching.js.org

Computing environment management with Conda/mamba

Conda

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

MENU nature methods

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

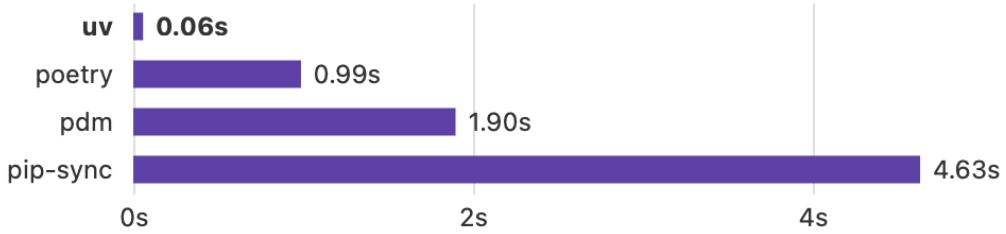
https://github.com/mamba-org/mamba

uv package management

uv

uv | pypi v0.6.6 | license MIT OR Apache-2.0 | python 3.8 | 3.9 | 3.10 | 3.11 | 3.12 | 3.13 | CI passing | Discord

An extremely fast Python package and project manager, written in Rust.



Installing Trio's dependencies with a warm cache.

Highlights

- 🚀 A single tool to replace `pip`, `pip-tools`, `pipx`, `poetry`, `pyenv`, `twine`, `virtualenv`, and more.
- ⚡ 10-100x faster than `pip`.
- 🗂️ Provides comprehensive project management, with a universal lockfile.

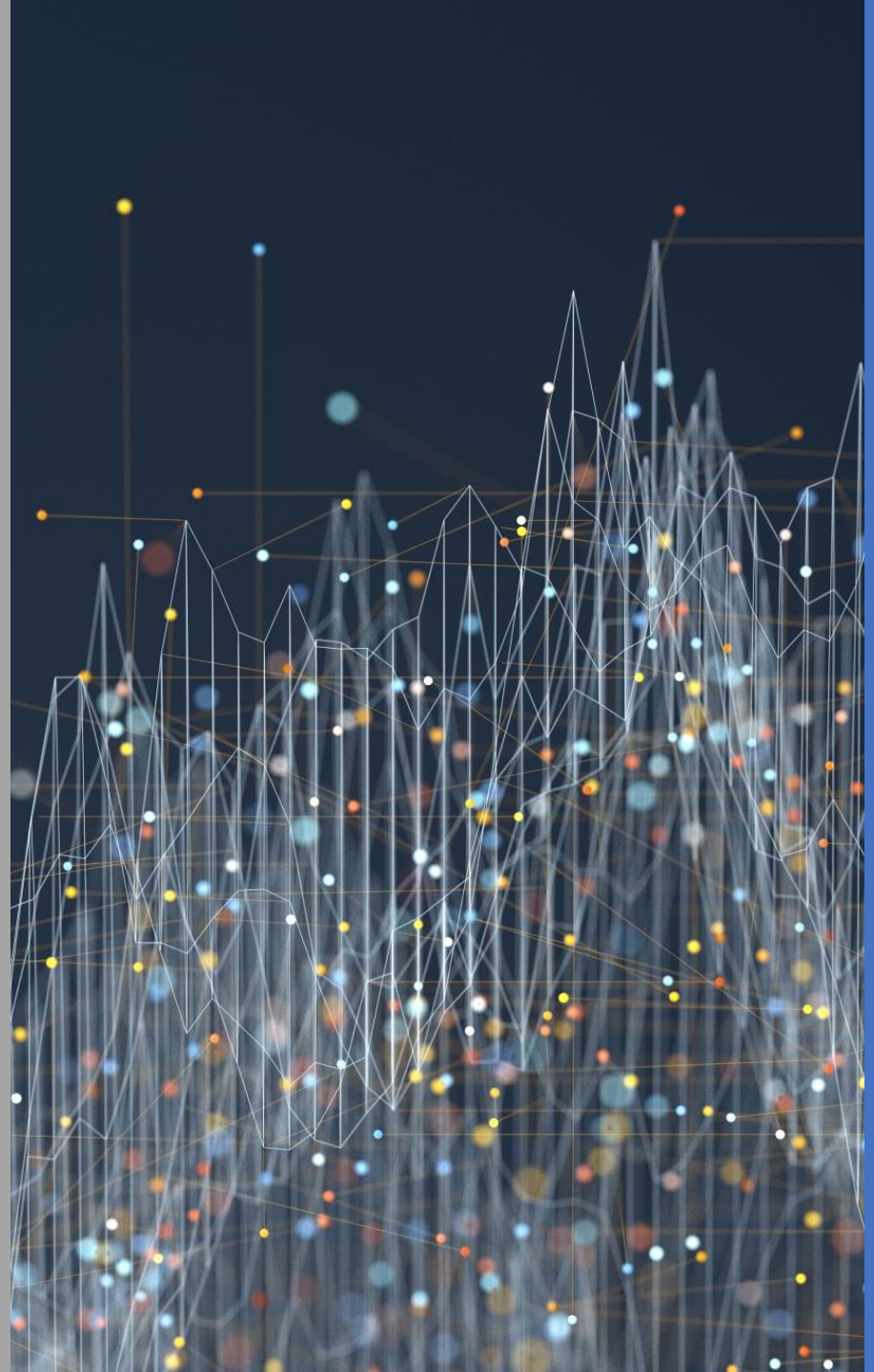
https://github.com/astral-sh/uv

renv

- The R package renv helps you to set up and restore project specific local environment
- Create a private R library with renv::int(). The project will now always rely on the local library
- Update a library with renv::snapshot()
- Restore a library with renv::restore()

How to ensure reproducibility

- Versioning of the data
- Naming files and Project organization
- Versioning of code: Git version control
- Versioning of packages: uv, conda, renv
- Versioning of operating systems: Containers (docker, singularity, biocontainers https://biocontainers.pro/)
- Use Jupyter/R Notebook, Quarto literal programming
- Clean code (functional programming)



Clean code with functional programming and R packages

Better code with functional programming

Functional programming

R, at its heart, is a functional programming (FP) language. This means that it provides many tools for the creation and manipulation of functions. In particular, R has what's known as first class functions. You can do anything with functions that you can do with vectors: you can assign them to variables, store them in lists, pass them as arguments to other functions, create them inside functions, and even return them as the result of a function.

http://adv-r.had.co.nz/Functional-programming.html
https://adv-r.hadley.nz/fp.html

Functional programming use purrr::map()

https://purrr.tidyverse.org

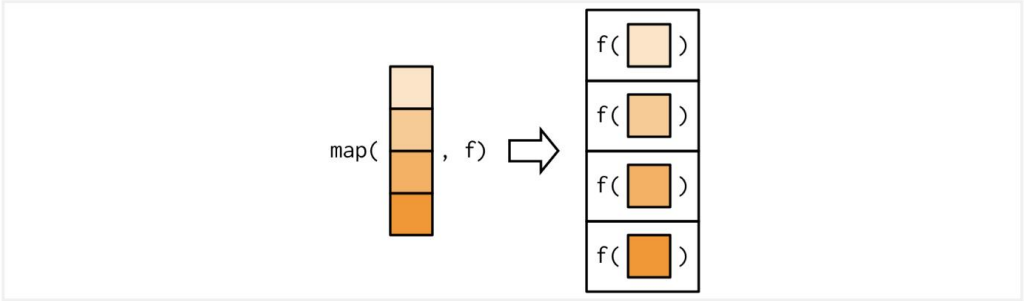
9.2 My first functional: `map()`

The most fundamental functional is `purrr::map()` [53]. It takes a vector and a function, calls the function once for each element of the vector, and returns the results in a list. In other words, `map(1:3, f)` is equivalent to `list(f(1), f(2), f(3))`.

```r
triple <- function(x) x * 3
map(1:3, triple)
#> [[1]]
#> [1] 3
#>
#> [[2]]
#> [1] 6
#>
#> [[3]]
#> [1] 9
```

Copy

Or, graphically:

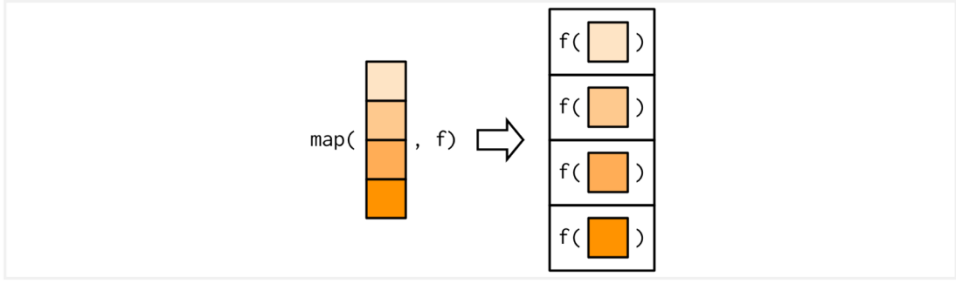


https://adv-r.hadley.nz/functionals.html

Functional programming (do not repeat yourself)

```
ht_list2 =
  partition_hp2 +
  EnrichedHeatmap(mat5, pos_line = FALSE,  column_title = "ER", row_title_rot = 0, name = "ER",
                  col= col_fun5,
                  top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = 2:4), ylim=c(0, 1))),
                  show_row_dend = FALSE) +
  EnrichedHeatmap(mat6, pos_line = FALSE,  column_title = "ER Capi", row_title_rot = 0, name = "ER Capi",
                  col= col_fun6,
                  top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = 2:4), ylim=c(0, 1))),
                  show_row_dend = FALSE) +
  EnrichedHeatmap(mat7, pos_line = FALSE,  column_title = "ER Ful", row_title_rot = 0, name = "ER Ful",
                  col= col_fun7,
                  top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = 2:4), ylim=c(0, 1))),
                  show_row_dend = FALSE) +
  EnrichedHeatmap(mat8, pos_line = FALSE,  column_title = "ER CF", row_title_rot = 0, name = "ER CF",
                  col= col_fun8,
                  top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = 2:4), ylim=c(0, 1))),
                  show_row_dend = FALSE) +
  EnrichedHeatmap(mat1, pos_line = FALSE,  column_title = "ATAC", row_title_rot = 0, name = "ATAC",
                  col= col_fun1,
                  top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = 2:4), ylim=c(0, 2.5))),
                  show_row_dend = FALSE) +
  EnrichedHeatmap(mat2, pos_line = FALSE,  column_title = "ATAC Capi", row_title_rot = 0, name = "ATAC Capi",
                  col= col_fun2,
                  top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = 2:4), ylim=c(0, 2.5))),
                  show_row_dend = FALSE) +
  EnrichedHeatmap(mat3, pos_line = FALSE,  column_title = "ATAC Ful", row_title_rot = 0, name = "ATAC Ful",
                  col= col_fun3,
                  top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = 2:4), ylim=c(0, 2.5))),
                  show_row_dend = FALSE) +
  EnrichedHeatmap(mat4, pos_line = FALSE,  column_title = "ATAC CF", row_title_rot = 0, name = "ATAC CF",
                  col= col_fun4,
                  top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = 2:4), ylim=c(0, 2.5))),
                  show_row_dend = FALSE)
```

```
```{r}
make_single_hp<- function(mat, sample_name, color_map, top_y_limit, kmeans_color){
  hp<- EnrichedHeatmap(mat, pos_line = FALSE,
                       column_title = sample_name,
                       row_title_rot = 0,
                       name = sample_name,
                       col= color_map,
                       top_annotation = HeatmapAnnotation(enriched = anno_enriched(gp = gpar(col = kmeans_color),
                                                                                   ylim= top_y_limit)),
                       show_row_dend = FALSE)
  return(hp)
}


H3K4me1_hps<- purrr::map2(H3K4me1_mats, names(H3K4me1_mats),
                          ~make_single_hp(mat = .x, sample_name = .y,
                                          color_map = col_fun2,
                                          top_y_limit = c(0, 0.6),
                                          kmeans_color = 2:4))

KMT2D_hps<- purrr::map2(KMT2D_mats, names(KMT2D_mats),
                        ~make_single_hp(mat = .x, sample_name = .y,
                                        color_map = col_fun1,
                                        top_y_limit = c(0, 0.8),
                                        kmeans_color = 2:4))

ht_list<-  purrr::reduce(H3K4me1_hps, `+`) + purrr::reduce(KMT2D_hps, `+`)
```


```

Use ChatGPT to refactor your code

```
### CCA

```{r}

# covariance matrix
Sigma_XY<- (1 / (min(ncol(centered_CCLE_cpm), ncol(centered_TCGA_cpm)) - 1)) * t(centered_CCLE_cpm) %*%
centered_TCGA_cpm

dim(Sigma_XY)

# Perform SVD
k <- 20  # Number of CCA dimensions
cca_svd <- irlba::irlba(Sigma_XY, nu=k, nv=k)

# Get canonical variates and correlations
canonical_variates_CCLE <- cca_svd$u  # CCLE canonical variates
canonical_variates_TCGA <- cca_svd$v  # TCGA canonical variates

# cca_svd$d contains the singular values
canonical_cors <- cca_svd$d  # Canonical correlations

range(canonical_cors)

# normalize it so the first CCA dimension has a correlation close to 1
canonical_cors<- cca_svd$d / sum(cca_svd$d)

barplot(canonical_cors,
        main="Canonical Correlations",
        xlab="Canonical Dimension",
        ylab="Correlation",
        col="lightblue")
```
```

```
# Function to align a query dataset to a reference dataset using CCA and perform label transfer
align_cca <- function(reference_data, query_data, reference_labels, k_cca = 20, k_mnn = 10) {
  # Normalize and scale
  centered_reference <- t(scale(t(reference_data), center = TRUE, scale = TRUE))
  centered_query <- t(scale(t(query_data), center = TRUE, scale = TRUE))

  # Covariance matrix
  sigma_xy <- (1 / (min(ncol(centered_query), ncol(centered_reference)) - 1)) *
              t(centered_query) %*% centered_reference

  # Perform SVD
  cca_svd <- irlba::irlba(sigma_xy, nu = k_cca, nv = k_cca)

  # Get canonical variates
  canonical_variates_query <- l2_normalize(cca_svd$u)  # Query canonical variates
  canonical_variates_reference <- l2_normalize(cca_svd$v)  # Reference canonical variates

  # Set row names to sample names and column names based on selected dimensions
  rownames(canonical_variates_query) <- colnames(query_data)
  colnames(canonical_variates_query) <- paste0("CCA", seq_len(k_cca))

  rownames(canonical_variates_reference) <- colnames(reference_data)
  colnames(canonical_variates_reference) <- paste0("CCA", seq_len(k_cca))

  # Find Nearest Neighbors using RANN
  nn_indices_reference <- nn2(data = canonical_variates_reference, query = canonical_variates_query, k =
k_mnn)$nn.idx
  nn_indices_query <- nn2(data = canonical_variates_query, query = canonical_variates_reference, k =
k_mnn)$nn.idx

  # Identify Mutual Nearest Neighbors
  mnn_indices <- find_mnn(nn_indices_reference, nn_indices_query, nrow(canonical_variates_query))

  # Perform Label Transfer
  label_result <- label_transfer(reference_labels, mnn_indices, nn_indices_reference)

  # Return the results
```