

Tools and tricks for a data scientist

03/09/2020 Ming (Tommy) Tang

You Retweeted

 **Jason Williams** @JasonWilliamsNY · 10h
Bioinformaticians rebranding themselves as Data Scientists.

 **Miss Distance (Kia☆)** @alt_kia · Mar 7
seems Legit
[Show this thread](#)



3 18 112

Oh-my-zsh!

- <https://ohmyz.sh/>

```
→ github_repos cd pyflow_scATACseq
→ pyflow_scATACseq git:(motif) ✘ ls README.md config.yaml
README.md           config.yaml          pyflow-scATACseq.sh*  samples.json
Snakefile           data/                rulegraph.png        scripts/
cluster.json        meta.txt            sample2json.py
```

<https://divingintogeneticsandgenomics.rbind.io/post/set-up-my-new-mac-laptop/>

Mosh: mobile shell

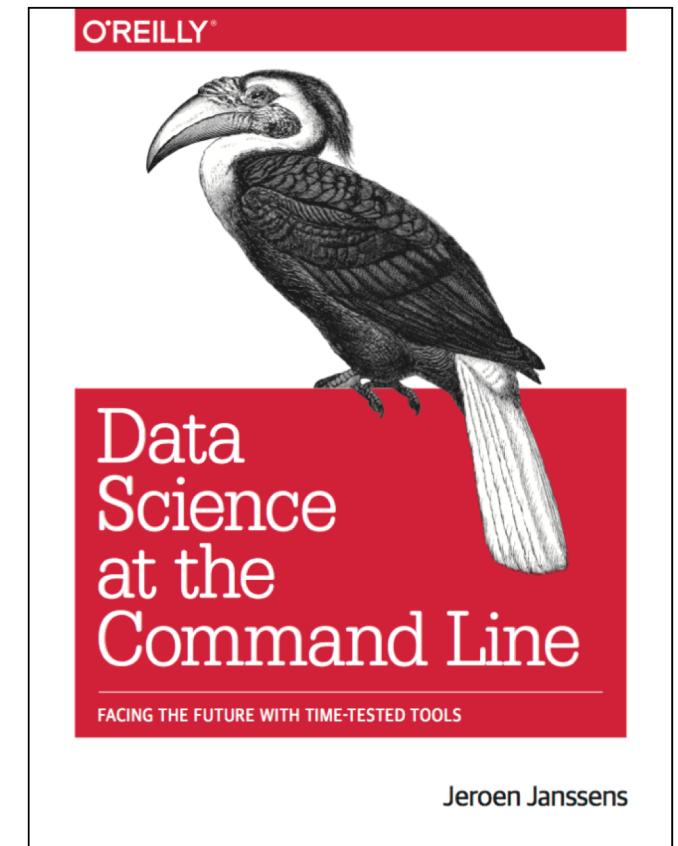
- <https://mosh.org/>
- Mosh + screen/tmux to keep your session persistent.

csvkit

- <https://www.datascienceatthecommandline.com/>

```
cd /n/holyscratch01/informatics/mtang
```

```
cat mtcars.csv | csvless -S  
cat mtcars.csv | head | csvless -S  
csvcut -n mtcars.csv
```



body

- <https://github.com/jeroenjanssens/data-science-at-the-command-line/blob/master/tools/body>

Executable File | 15 lines (15 sloc) | 472 Bytes Raw

```
1 #!/usr/bin/env bash
2 #
3 # body: apply expression to all but the first line.
4 # Use multiple times in case the header spans more than one line.
5 #
6 # Example usage:
7 # $ seq 10 | header -a 'values' | body sort -nr
8 # $ seq 10 | header -a 'multi\nline\nheader' | body body body sort -nr
9 #
10 # From: http://unix.stackexchange.com/a/11859
11 #
12 # See also: header (https://github.com/jeroenjanssens/command-line-tools-for-data-science)
13 IFS= read -r header
14 printf '%s\n' "$header"
15 eval $@
```

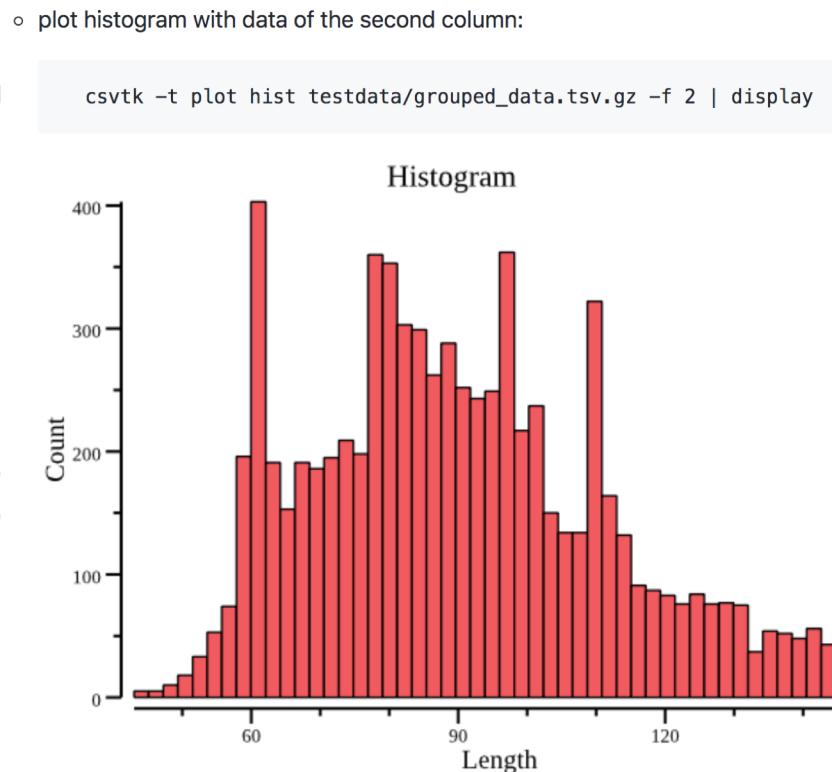
Cat myfile.txt | body grep “pattern”

Will retain the header

cat mtcars.csv | body grep Ford | csvless -S

csvtk

- <https://github.com/shenwei356/csvtk>
- E.g. cut out columns based on column names in another file.
 - plot histogram with data of the second column:
- csvtk cut -f \$(paste -s -d, columns.txt) mtcars.csv
- Unix cut can not arrange column orders,
- I usually use awk. Csvtk can
- Other tools:
- <https://github.com/crazyhottommy/getting-started-and-resources#do-not-give-me-excel-files>



GNU parallel

```
Download for all
```

```
```bash
cat 2020-02-24_sophia_basespace_run_info.txt | awk
'{print$2}' | sed '1d' | grep -v 117072956 | grep -v
11601489 > bs_cli/run_ids.txt
```

```
cd bs_cli
cat run_ids.txt | parallel -j 24 'echo bs download run -
c default -i {} -o {} --log={}.log'
```

```
cat run_ids.txt | parallel -j 24 'echo bs download run -
c default -i {} -o {} --log={}.log'
```

```
...
```

# Most frequently used...

- 1. `readlink -e`
- 2. `realpath`
- 3. `less -S`
- 4. `cat -A` show hidden characters e.g. `^M`, `^I`,
- 5. `dos2unix`

# One-liners

- <https://github.com/crazyhottommy/bioinformatics-one-liners>

The screenshot shows the GitHub repository page for 'crazyhottommy / bioinformatics-one-liners'. The page includes the repository name, a summary of activity (0 issues, 0 pull requests), integration links (ZenHub, Actions, Projects, Wiki, Security, Insights, Settings), and a detailed description of the repository's purpose.

crazyhottommy / **bioinformatics-one-liners** ≡

Unwatch ▼ 10 ⭐ Star 244 Fork 59

Code Issues 0 Pull requests 0 ZenHub Actions Projects 0 Wiki Security Insights Settings

Bioinformatics one liners from Ming Tang Edit

# Brenamer: rename your files without a mess

- <https://github.com/shenwei356/brenamer>
- Written in go, download the binary, ready to use.
- Regular expression
- undo last -u
- Dry run -d
- only renaming specific paths via include filters :
- brenamer -p ":" -r "-" -f ".htm\$" -f ".html\$"
- using capture variables, e.g., \$1, \$2...
- brenamer -p "(m)" -r "\\$1\\$1"

# rmate editing remote files (I only know how to quit vim)

- <https://divingintogeneticsandgenomics.rbind.io/post/open-files-on-remote-with-sublime-by-ssh/>

on remote machine, install `rmate`

```
ssh bio1
curl -o ~/bin/rmate https://raw.githubusercontent.com/aurora/rmate/master/rmate
chmod u+x bin/rmate
```

on your local computer, install `RemoteSubl`

on your **local** computer, open `sublime`, click `tools` → `Command Palette` → type `Package control:Install Package` → type `RemoteSubl` to install.

**change your ssh config file**

add `RemoteForward 52698 localhost:52698` to your `~/.ssh/config` file.

Now, ssh to remote, and you can do `rmate my.txt` in your remote and open sublime in your local machine.

# ncdu

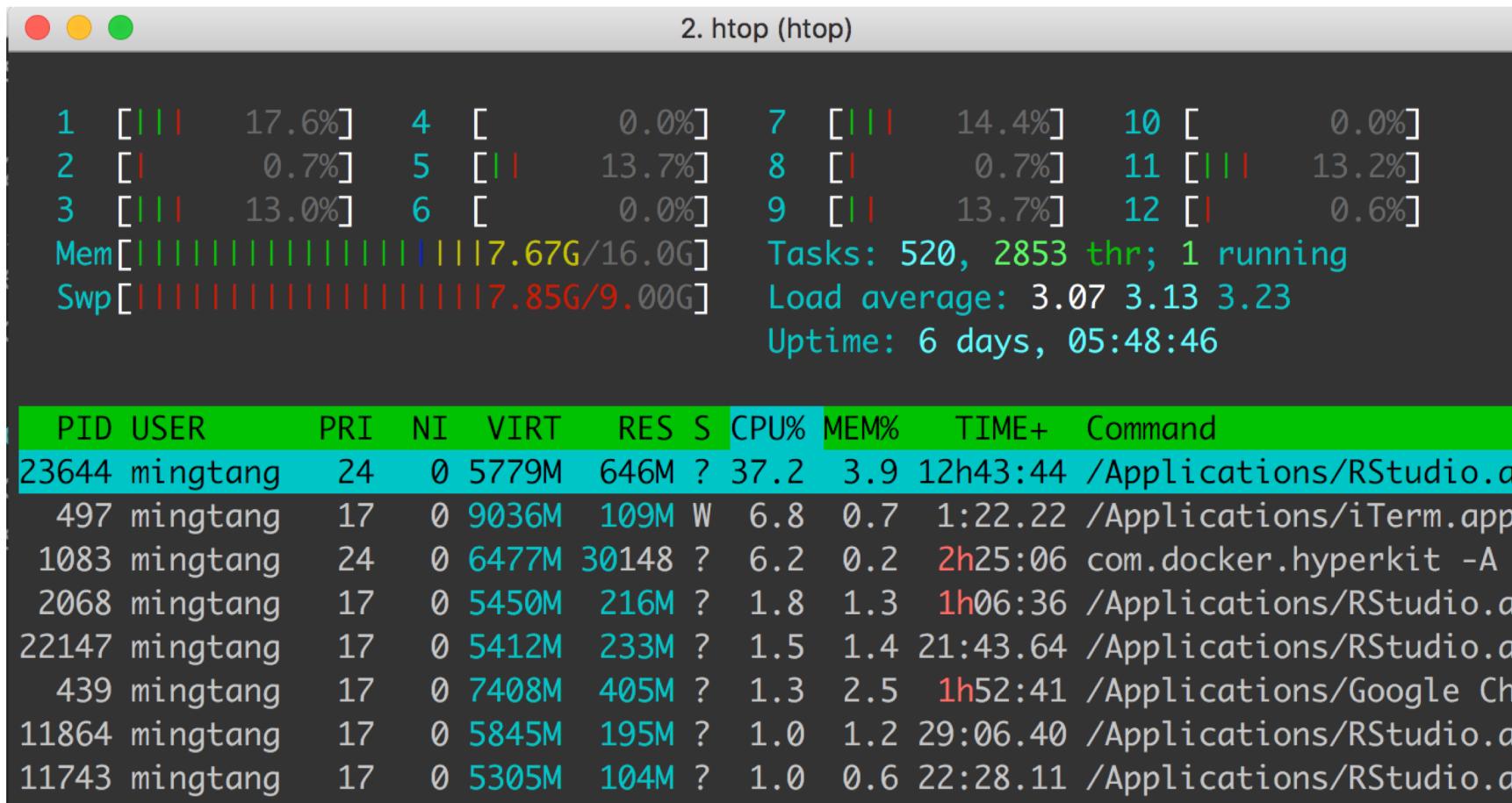
<https://anaconda.org/coecms/ncdu>

- Ncdu, acronym of **NCurses Disk Usage**, is a curses-based version of the well-known ‘du’ command. It provides a fast way to see what directories are using the disk space.

```
ncdu 1.12 ~ Use the arrow keys to navigate, press ? for help
--- /Users/mingtang/projects ---
22.8 GiB [#####] /brandon_scATAC
13.8 GiB [#####] /brandon_galanin_hypothalamus_sNucSeq
13.1 GiB [#####] /EvaluateSingleCellClustering
6.9 GiB [##] /sophia_hypothalamus_sNucSeq
5.6 GiB [##] /playground
2.4 GiB [#] /dj_marmoset_single_cell
1.8 GiB [] /Hunain_Mll4
573.9 MiB [] /dj_marmoset_single_cell_packrat
268.4 MiB [] /modern_stat_modern_bio
42.3 MiB [] /STATE80
13.4 MiB [] /dulaclab_random_help
2.0 MiB [] fusionlist_20160124.txt
72.0 KiB [] /kallisto-scATAC
44.0 KiB [] /deeplearningwithr
28.0 KiB [] .DS_Store
12.0 KiB [] scclusteval_manuscript.docx
4.0 KiB [] ~$clusteval_manuscript.docx
```

# higher top: htop

<https://hisham.hm/htop/>



# Dat:

- peer-to-peer sharing & live synchronization of files via command line <https://dat.foundation>.
- npm install -g dat



# Notion App for to do list and many more

The screenshot shows the left sidebar of a Notion database. At the top is a user profile icon for 'M Ming'. Below it are four quick access links: 'Quick Find', 'All Updates', and 'Settings & Members'. The main list contains 21 entries, each represented by a small document icon followed by a date and 'todo list'. The entries are: '08/19/2019 todo list', '08/26/2019 todo list', '09/09/2019 todo list', '09/16/2019 todo list', '09/23/2019 todo list', '09/29/2019 todo list', '10/07/2019 todo list', '10/15/2019 todo list', '10/21/2019 todo list', '11/04/2019 todo list', '11/11/2019 todo list', '11/18/2019 todo list', '12/02/2019 todo list', '12/09/2019 todo list', '12/16/2019 todo list', '01/02/2020 todo list', '01/06/2020 to do list', '01/13/2020 todo list', '01/27/2020 todo list', '02/03/2020 todo list', '02/10/2020 todo list' (which is highlighted with a gray background), '02/17/2020 todo list', and '02/24/2020 todo list'.

## 02/10/2020 todo list

debugging filtering seATACseq

extending Tdtomato, still not many cells with counts?! after cellranger

extended Tdtomato with Mapsembler3, (3 days)

cellranger 154 cells express Tdtomato (3days again...)

I then tried kallisto + bustools, something is strange. The fastq from brandon v3 fastqs are 16bp cell barcode + 10 bp UMI, it should be 12 bp UMI for v3 and 10 bp for V2?!!

That's why kallisto keeps failing me, because I specify -x 10xv3.

+  If I use -x 10xv2, it run, but the cell barcode are all wrong.

I had to use -x 0,0,16:0,16,26:1,0,0 -w 3M-february-2018.txt to specify manually. It finally worked...

222 cells express Tdtomato. (more counts in general for kb-python)

I then wrote to the Allen Institute and got the DNA sequence.

Mapsembler did a great job, but somehow repeat 30bp, the rest several hundred are the same as in the vector DNA sequence.

# Hackmd for taking notes

- <https://hackmd.io/>

The screenshot shows the HackMD interface. At the top, there's a toolbar with various icons for editing and sharing. Below it is a header bar with the title 'HackMD' and a user profile. The main area contains a note titled 'downloading for sophia'. The note includes several code snippets in a monospaced font, some of which are highlighted in orange. One snippet shows a command to download data from BaseSpace. Another section is titled 'use bs-cli instead' and provides a link to the documentation. At the bottom, there's a footer with a 'Download for all' button.

```
downloading for sophia
...
sudo python BaseSpaceRunDownloader_v2.py -r 117072956 -a 482e50b8852d4609a414e5badad7debe > 117072956.log 2>&1

sudo python BaseSpaceRunDownloader_v2.py -r 116014899 -a 482e50b8852d4609a414e5badad7debe
```

### use bs-cli instead
https://developer.basespace.illumina.com/docs/content/documentation/cli/cli-overview

```bash
bs download run -c default -i 117072956 -o 117072956 --log=117072956.log

bs download run -c default -i 116014899 -o 116014899 --log=116014899.log
```

get all the run information, total 37 runs shared by Sophia.

bs -c default list run | tail -n+2 | body grep -i -e liang -e sub06 | se
```

Download for all

Take notes and maybe write it to a blog post.

Blogdown for blog posts

DNA CONFESSES DATA SPEAK

[Home](#) [Publications](#) [Posts](#) [Projects](#) [Talks & Teachings](#) [CV](#) [Contact](#)



About me

I am a computational biologist working on genomics, epigenomics and transcriptomics. I use R primarily for data wrangling and visualization in the [tidyverse](#) ecosystem; I use python for writing [Snakemake](#) workflows and reformatting data; I am a unix geek learning shell tricks almost every month; I care about reproducible research and open science.

I also have a great interest in promoting open science and reforming bioinformatics education. I frequently share my thoughts on [twitter](#) and tips in my [blog post](#). I am a certified instructor for the [carpentries](#).

Ming Tang

Bioinformatics Scientist

Harvard Faculty of Arts and Sciences
informatics

<https://divingintogeneticsandgenomics.rbind.io/post/hugo-academic-theme-blog-down-deployment-some-details/>

Workflowr to make website for teaching, sharing projects

- <https://github.com/jdblischak/workflowr>

scRNA-seq-workshop-Fall-2019

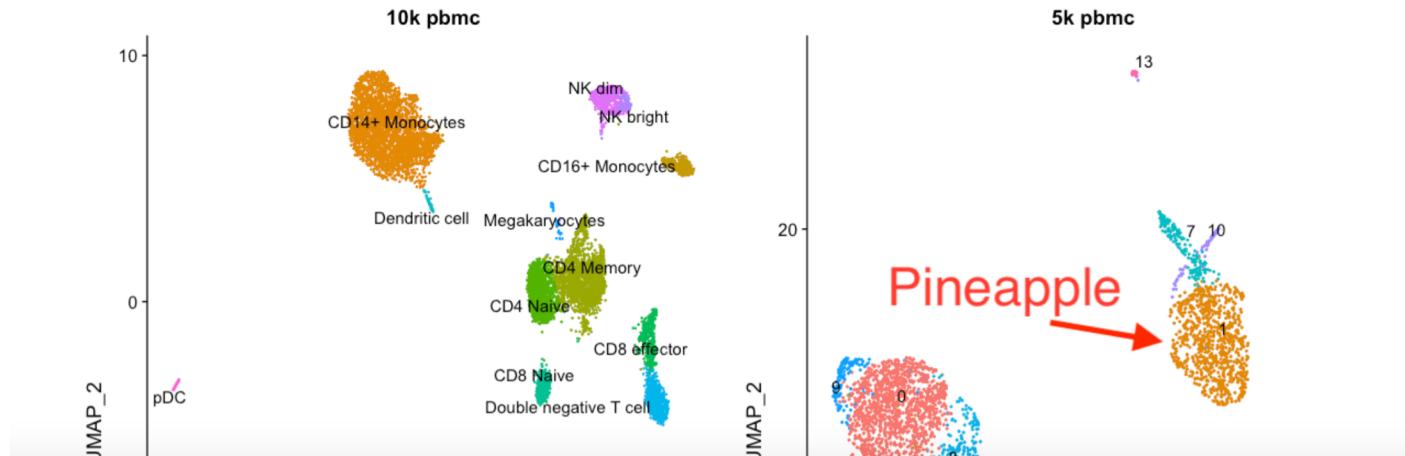
Home About Content ▾ License

Source code

Home



This is the main page of the [Harvard FAS informatics](#) scRNAseq workshop (part of the nanocourse) held from August 19th - 22nd.



<https://crazyhottommy.github.io/scRNA-seq-workshop-Fall-2019/>

Command line R utilities

- DocoptR
- <https://divingintogeneticsandgenomics.rbind.io/post/use-docopt-to-write-command-line-r-utilities/>
- Littler
- <http://dirk.eddelbuettel.com/code/littler.html>
- Funr
- <https://github.com/sahilseth/funr>

Rstudio R project

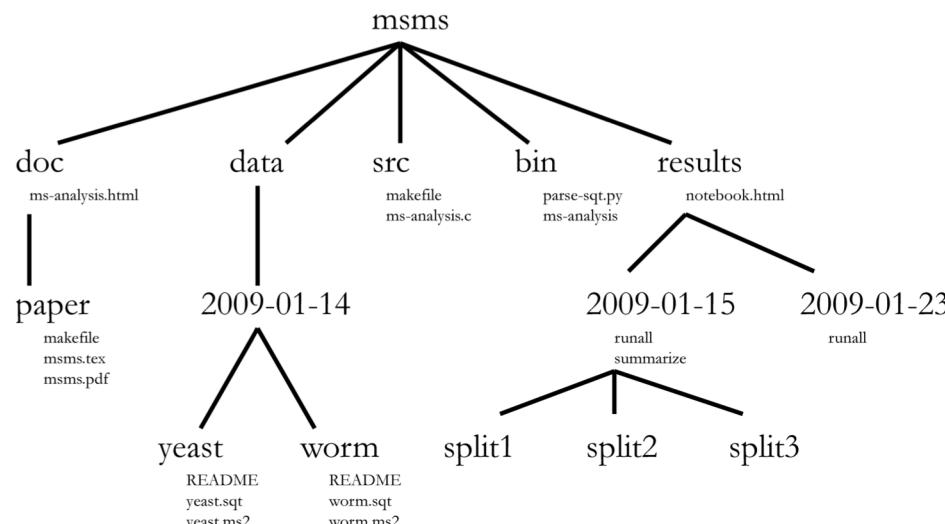
The screenshot shows the RStudio interface with the following components:

- Left Panel (Code Editor):** An R Markdown file titled "AC.Rmd" is open. The code is written in R, primarily using the Seurat package for scATAC-seq analysis. It includes sections for setting the title, loading libraries, querying an AnnotationHub for gene annotations, and exporting the results to a GTF file. It also handles peak files from Read10X_h5 and cellranger.
- Top Bar:** Shows tabs for "AC.Rmd", "2019-05-02_galanin_scATAC_TFI...", "2019-05-03_clustering_10k_pbmc...", and "20". There are also standard RStudio icons for file operations, search, and knit.
- Right Panel (Global Environment):** The "Environment" tab is selected. The message "Environment is empty" is displayed. Below it, the "Files" tab is active, showing a list of RMD files in the "scripts" directory of a project named "brandon_scATAC".
- Bottom Navigation:** Includes tabs for "Console", "Chunk 5", and "R Markdown".

 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. Plumley, Ben Waugh, Ethan P. White, Paul Wilson

here::here()

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

Works with Rproject

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 🔥.

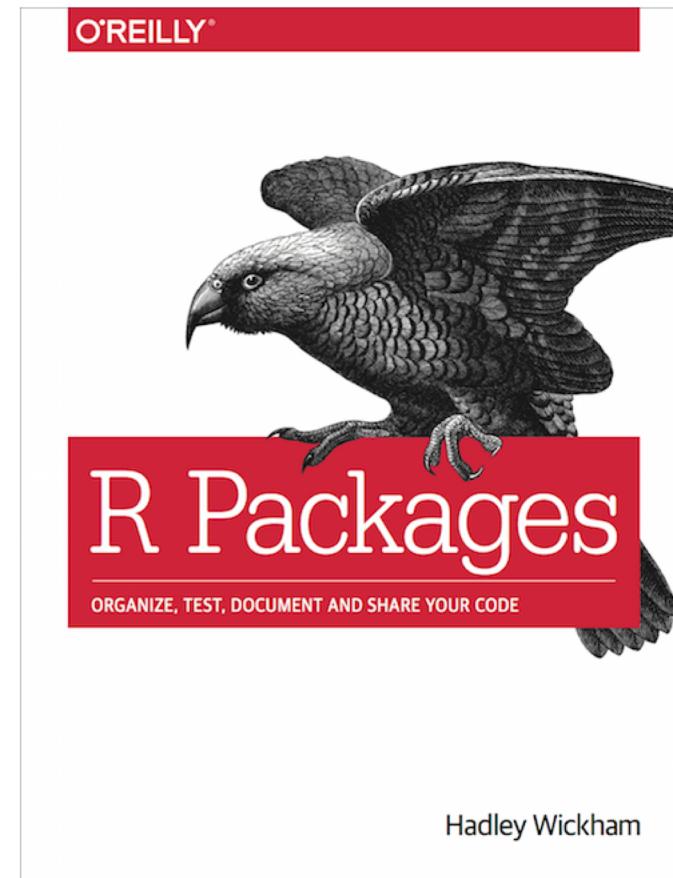
If the first line of your R script is

```
rm(list = ls())
```

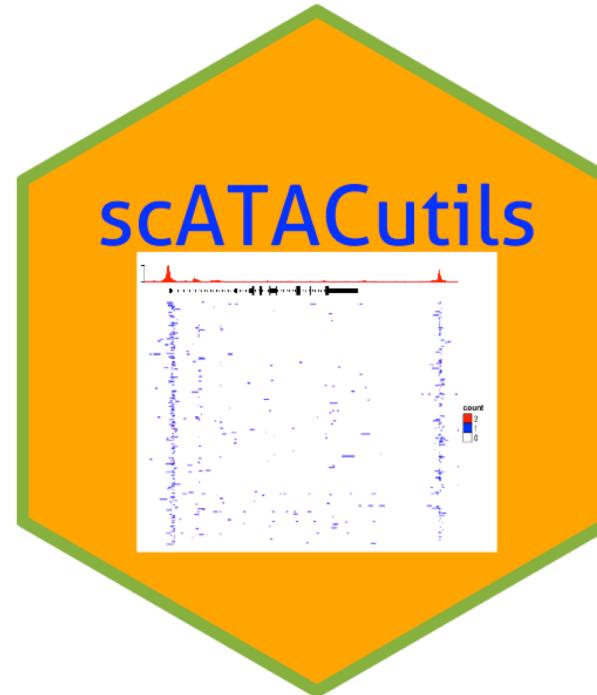
I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.

Making R packages

- <http://r-pkgs.had.co.nz/>



R packages



<https://github.com/crazyhottommy/scclusteval>
<https://github.com/crazyhottommy/scATACutils>

Docker + rstudio (Thanks Nathan!)

- Docker/singularity rocker image
- Ssh tunneling to connect to bioinfo1 (enjoy the 1 TB RAM!)

First, go to <https://www.rocker-project.org/images/> choose the image you want.

I use `tidyverse` heavily, so I downloaded the `tidyverse` image built upon
`Rstudio` image

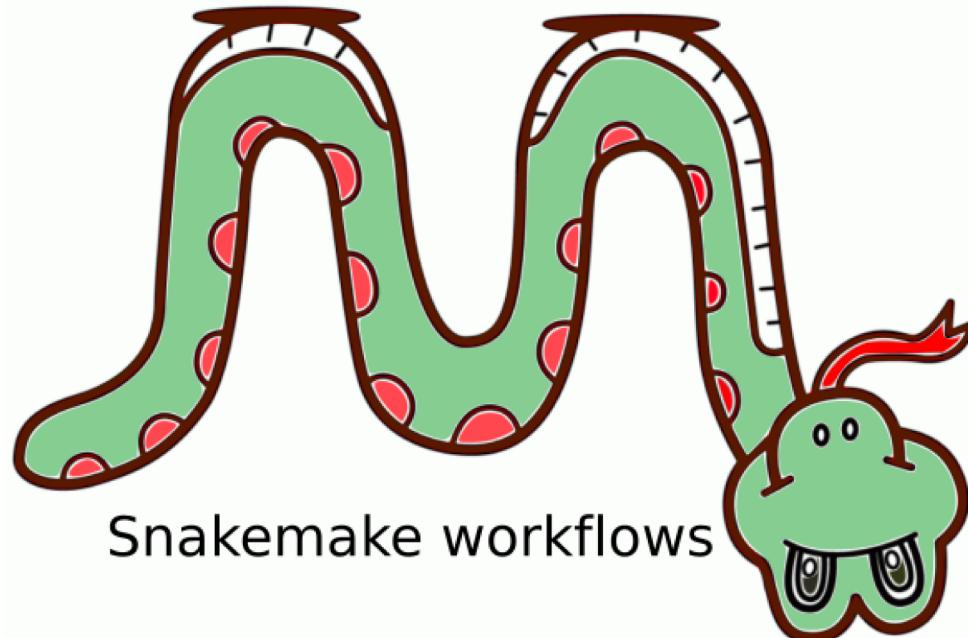
```
## ssh to remote HPC and pull the docker image by singularity
ssh biol
mkdir singularity-images; cd !$
singularity pull --name rstudio.simg docker://rocker/tidyverse:latest

# This example bind mounts the /project directory on the host into the Singularity container
# By default the only host file systems mounted within the container are $HOME, /tmp, /proc
# type in the password you want to set, make it more complicated than this dummy one
PASSWORD='xyz' singularity exec --bind=/project rstudio.simg rserver --auth-none=0 --au
```

- <https://divingintogeneticsandgenomics.rbind.io/post/run-rstudio-server-with-singularity-on-hpc/>

Snakemake for pipelines

- <https://snakemake.readthedocs.io/en/stable/>
- tutorials
- <https://github.com/ctb/2019-snakefile-ucdavis>
- <https://hackmd.io/jXwbvOyQTqWqpuWwrpByHQ?view>



Many workflow languages/engines

Awesome Pipeline

A curated list of awesome pipeline toolkits inspired by [Awesome Sysadmin](#)

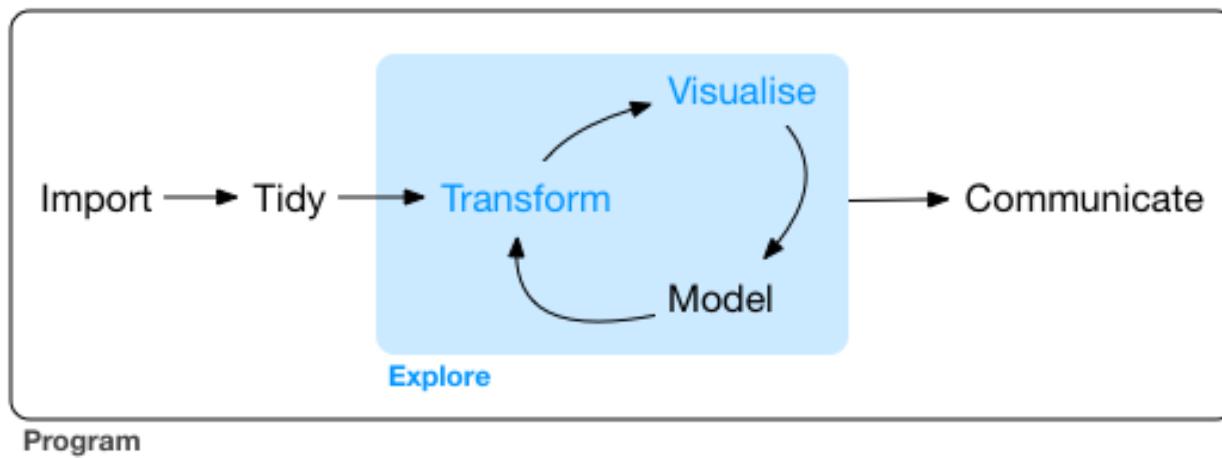
Pipeline frameworks & libraries

- [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.



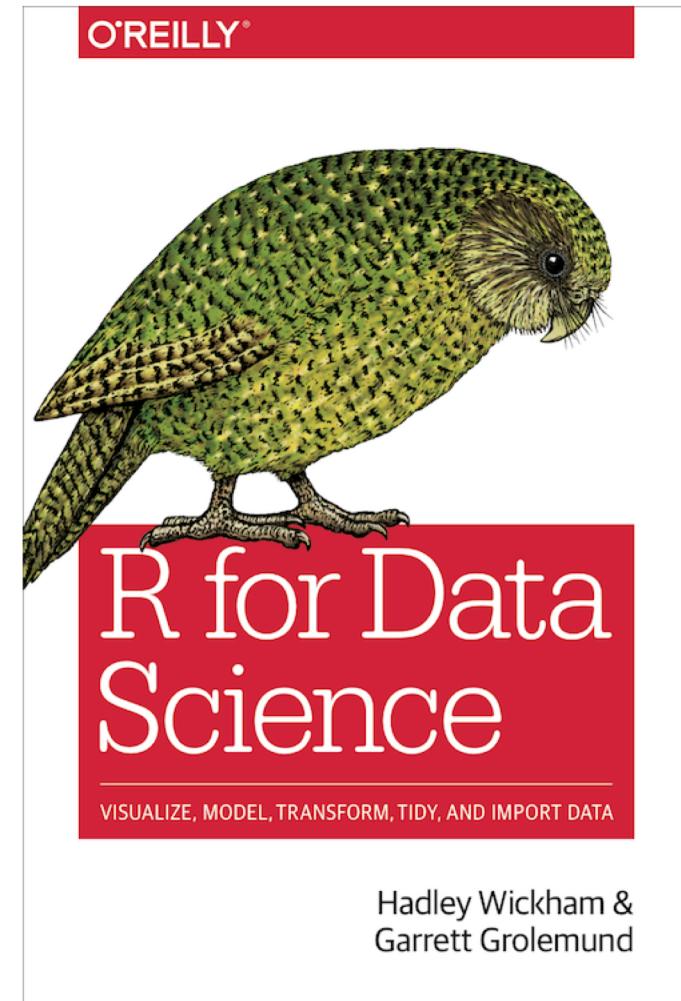
nextflow

Downstream analysis



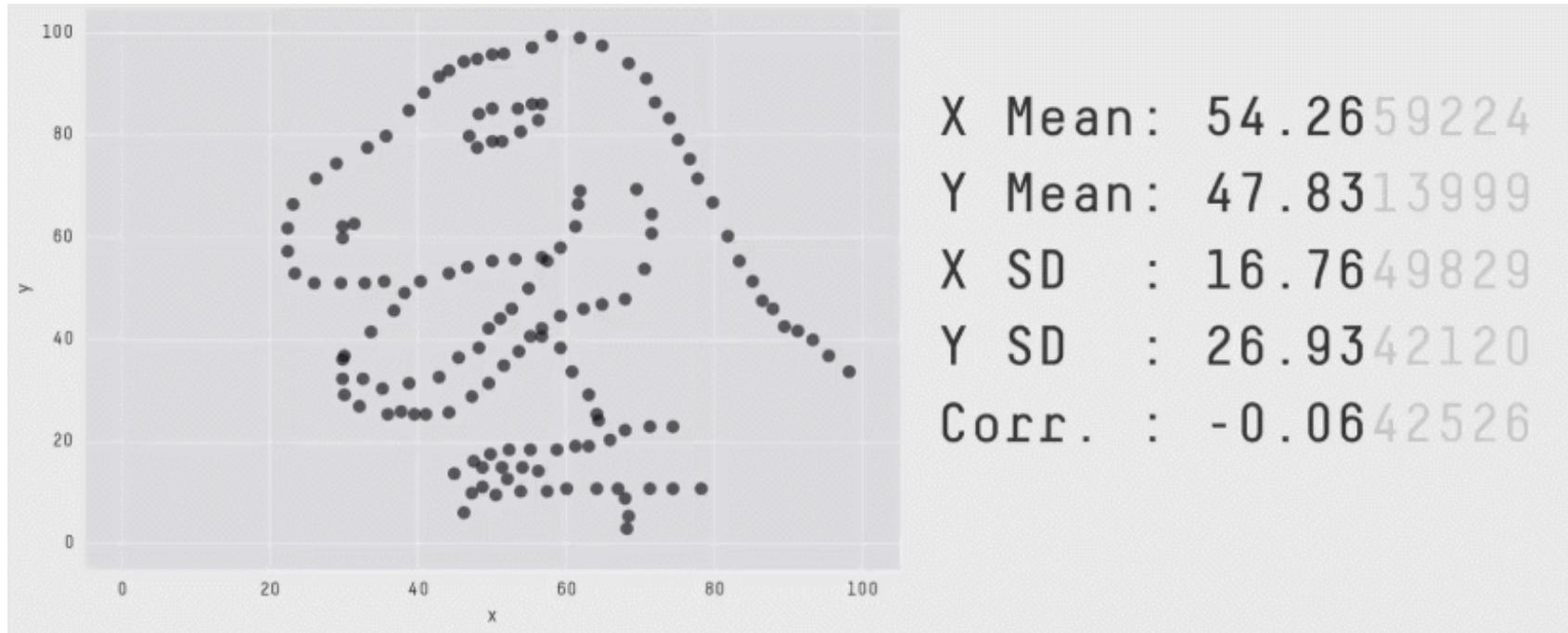
Tidying the data can take 80% of your time
Tidyverse

R for data science by Hadley Wickham & Garrett Grolemund
<http://r4ds.had.co.nz/>



Hadley Wickham &
Garrett Grolemund

Data visualization



<https://www.r-bloggers.com/the-datasaurus-dozen/>

One single suggestion

- Documentation! Documentation! And documentation!

One last suggestion: backup!

Backup by crontab

- <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
```

#rsync every Sunday 5am.

crontab file

crontab syntax

```
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
```

```
*   *   *   *   *           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)
```