

Tools and tricks for a data scientist

03/09/2020 Ming (Tommy) Tang



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

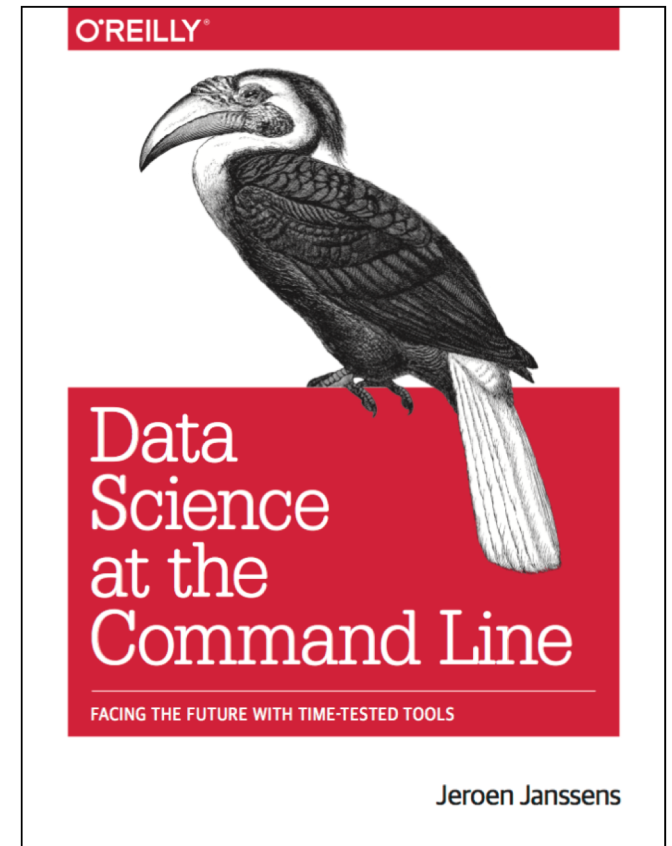
- <https://www.datascienceatthecommandline.com/>

```
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.**

- `csvtk cut -f $(paste -s -d, columns.txt) mtcars.csv`

- **Unix cut can not arrange column orders,**

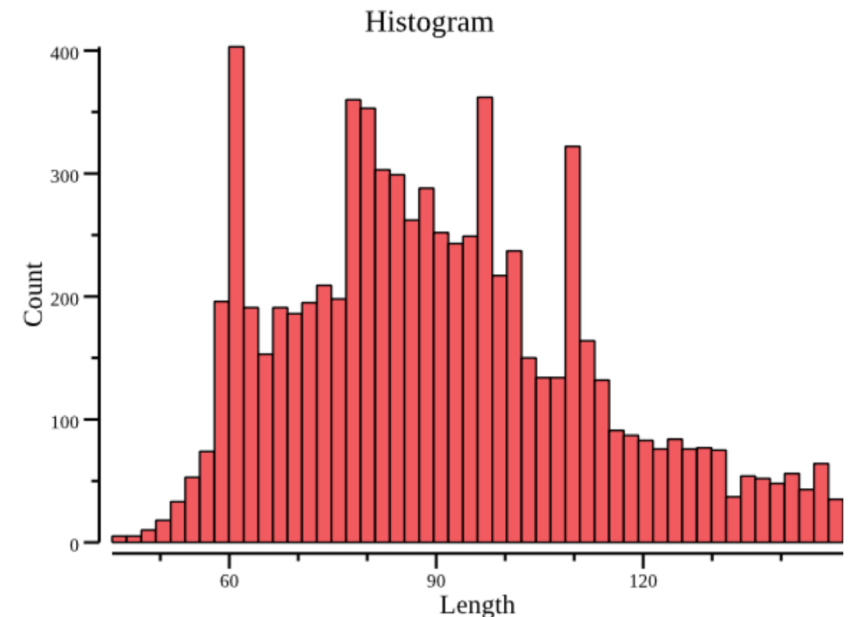
- **I usually use awk. Csvtk can**

- **Other tools:**

- <https://github.com/crazyhottommy/getting-star-tools-and-resources#do-not-give-me-excel-files>

◦ plot histogram with data of the second column:

```
csvtk -t plot hist testdata/grouped_data.tsv.gz -f 2 | display
```



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>

 crazyhottommy / bioinformatics-one-liners

 Unwatch ▼

10

★ Star

244

 Fork

59

<> Code

! Issues 0

 Pull requests 0

Z ZenHub

▶ Actions

 Projects 0

 Wiki

 Security

 Insights

⚙ Settings

Bioinformatics one liners from Ming Tang

Edit

Brename: rename your files without a mess

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

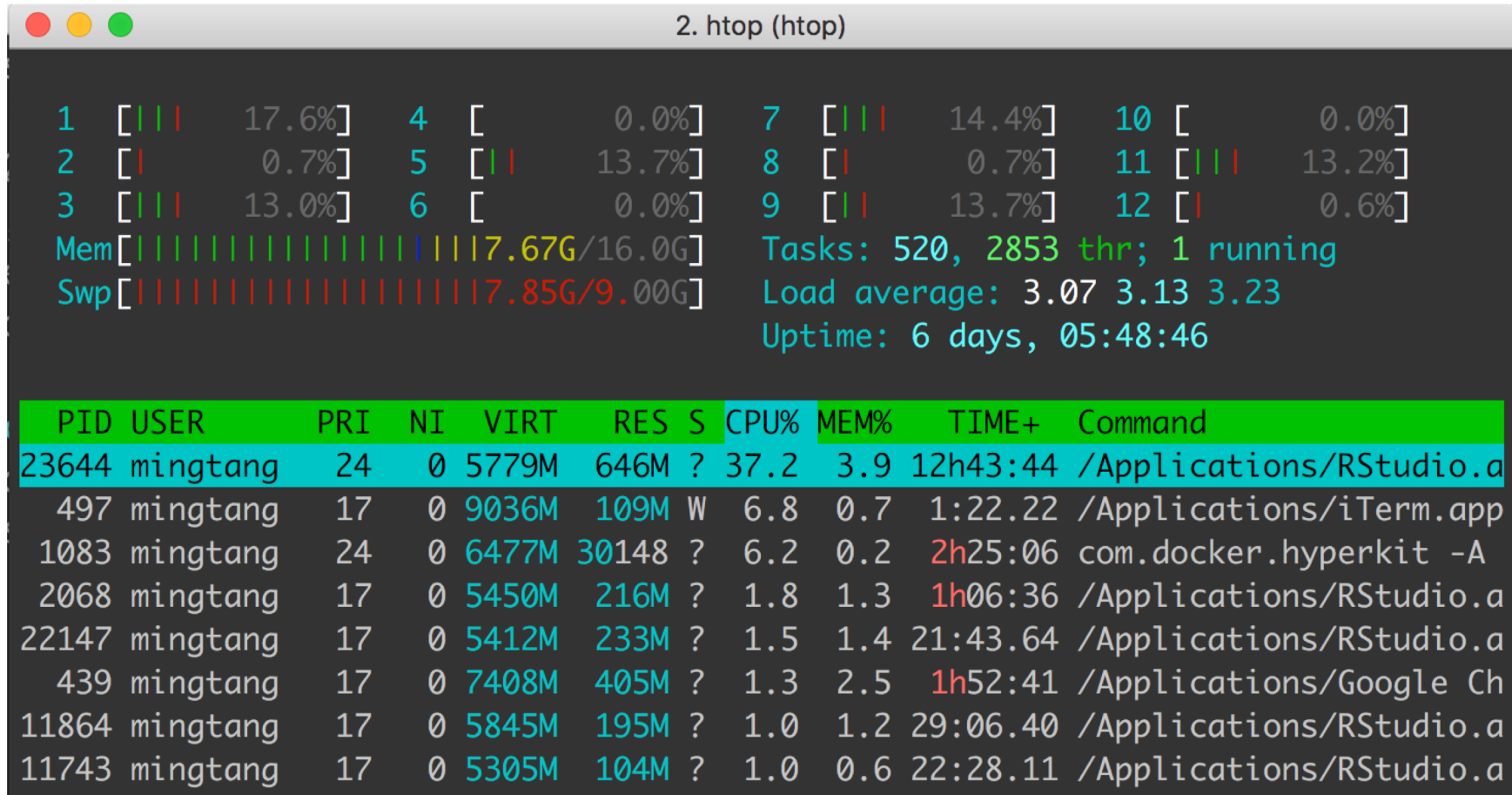
<https://anaconda.org/coecms/ncdu>

- Ncdu, acronym of **N**Curses **D**isk **U**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 [ ] /dulacrab_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
```


higer top: htop

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

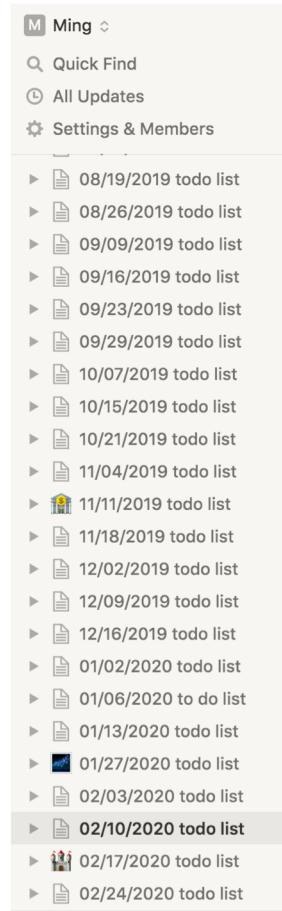


Dat:

- 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



02/10/2020 todo list

- ✓ debugging-filtering-scATACseq
- ✓ 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 a HackMD interface with a dark-themed editor on the left and a rendered view on the right. The editor contains a note with the following content:

```
1 # downloading for sophia
2
3
4 sudo python BaseSpaceRunDownloader_v2.py -r 117072956 -a 482e50b8852d4609a414e5badad7debe > 117072956.log 2>&1
5
6
7 sudo python BaseSpaceRunDownloader_v2.py -r 116014899 -a 482e50b8852d4609a414e5badad7debe
8
9
10 ### use bs-cli instead
11 https://developer.basespace.illumina.com/docs/content/documentation/cli/cli-overview
12
13 ``bash
14
15 bs download run -c default -i 117072956 -o 117072956 --log=117072956.log
16
17 bs download run -c default -i 116014899 -o 116014899 --log=116014899.log
18
19
20 get all the run information, total 37 runs shared by Sophia.
```

The rendered view on the right shows the same content with syntax highlighting and a link to the BaseSpace CLI documentation. It includes a 'Watch' button and a '1 ONLINE' indicator.

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](#)



Ming Tang

Bioinformatics Scientist

[Harvard Faculty of Arts and Sciences](#)
[informatics](#)

About me

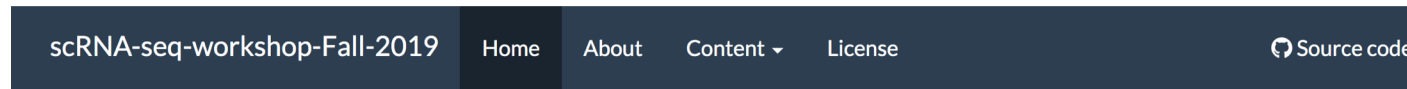
I am a computational biologist working on genomics, epigenomics and transcriptomics. I use R primary for data wrangling and visualization in the [tidyverse](#) ecosystem; I use python for writing [Snakemake](#) workflows and reformatting data; I am a unix geek learning shell tricks almost every month;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

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

Workflow to make website for teaching, sharing projects

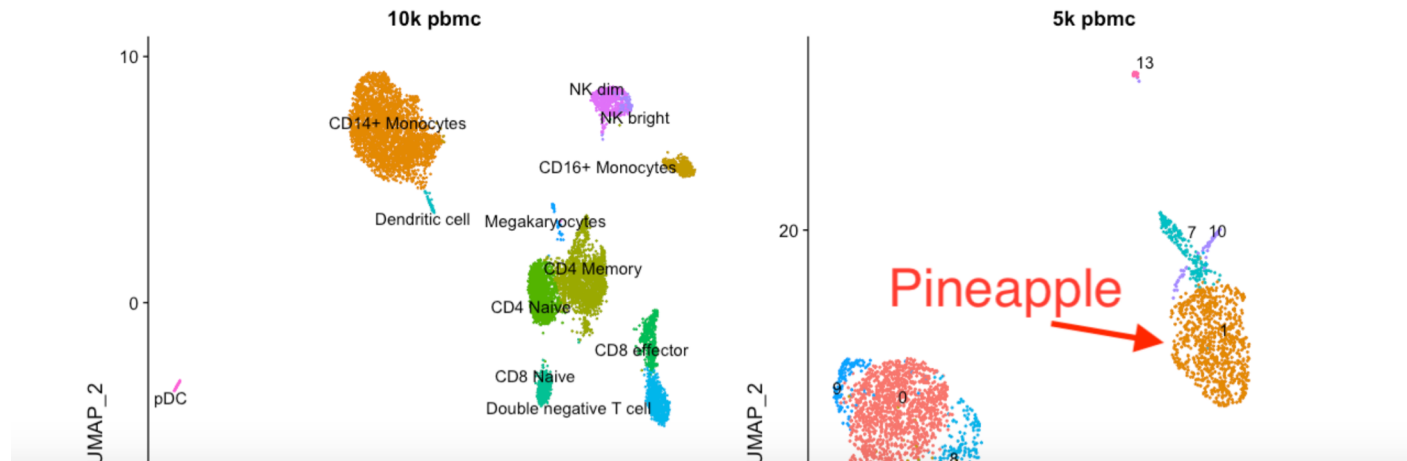
- <https://github.com/jdblischak/workflowr>



Home

workflowr ✓

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 utilites

- 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 displays the RStudio R project interface. The main editor window shows an R Markdown file with the following content:

```
1 ---
2 title: "2019-05-02_TF-IDF_scATAC"
3 output: html_document
4 ---
5 Brandon sequenced 3 flow cells for the galanin scATACseq data set
6 `cellranger-atac account` was run for all fastqs combined.
7
8 ```{r gene.activity.matrix}
9 library(Seurat)
10 # biocManager
11 library(AnnotationHub)
12 library(rtracklayer)
13 ah<- AnnotationHub()
14
15 ## need a gene annotation file for mm10
16 query(ah, c("gtf", "GRCm38"))
17
18 ## Mus_musculus.GRCm38.94.gtf
19 gtf<- ah[["AH64689"]]
20
21 export(gtf, "data/Mus_musculus.GRCm38.94.gtf", format = "gtf")
22
23 ## Read10X by data.dir not working, expect gz file... cellranger-atac 1.0 internally
24   uses cellranger <= 3.0
25
26 Father_peaks <- Read10X_h5( filename =
27   "data/galanin_scATAC/Fathers/filtered_peak_bc_matrix.h5")
28
29 VirginMale_peaks <- Read10X_h5( filename =
30   "data/galanin_scATAC/VirginMales/filtered_peak_bc_matrix.h5")
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

The environment pane on the right shows the Global Environment, which is empty. The file explorer on the bottom right shows the project structure, including a list of R Markdown files in the scripts directory.

The console at the bottom shows the execution of the R code, with the message "Chunk 5" displayed.

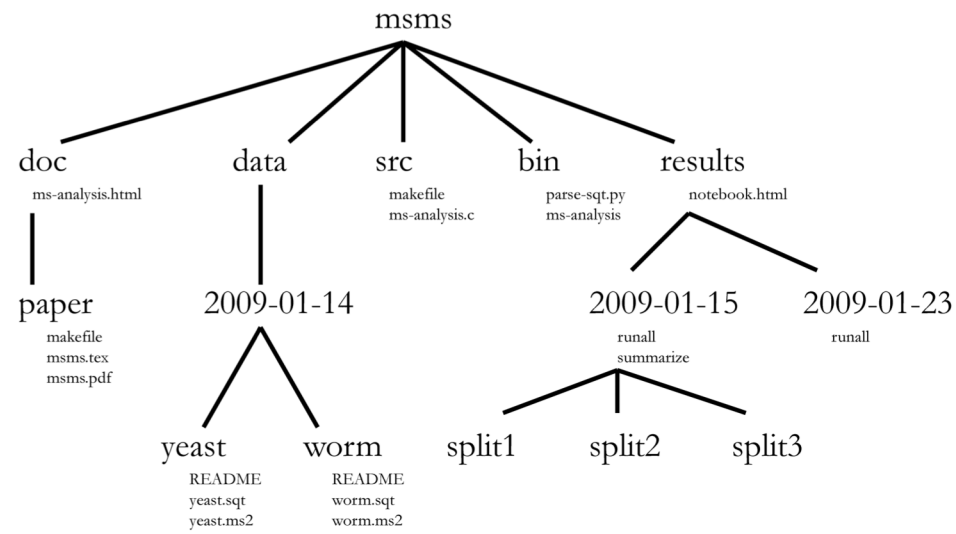
 OPEN ACCESS

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

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())
```

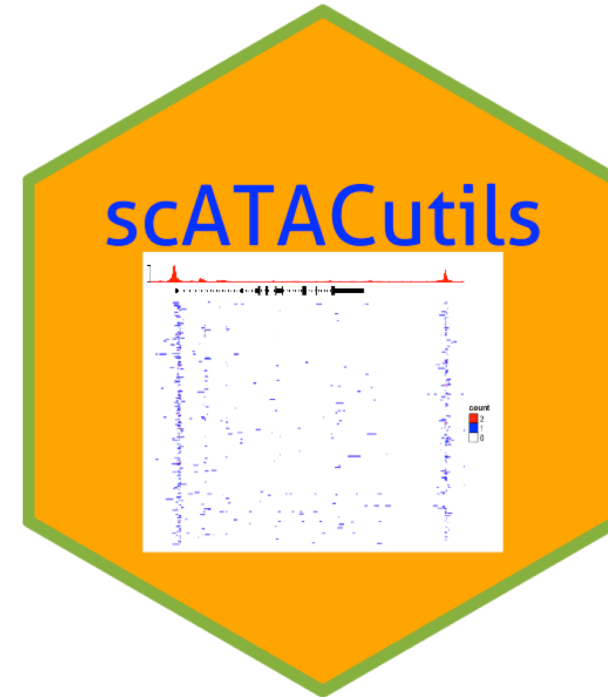
I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.

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

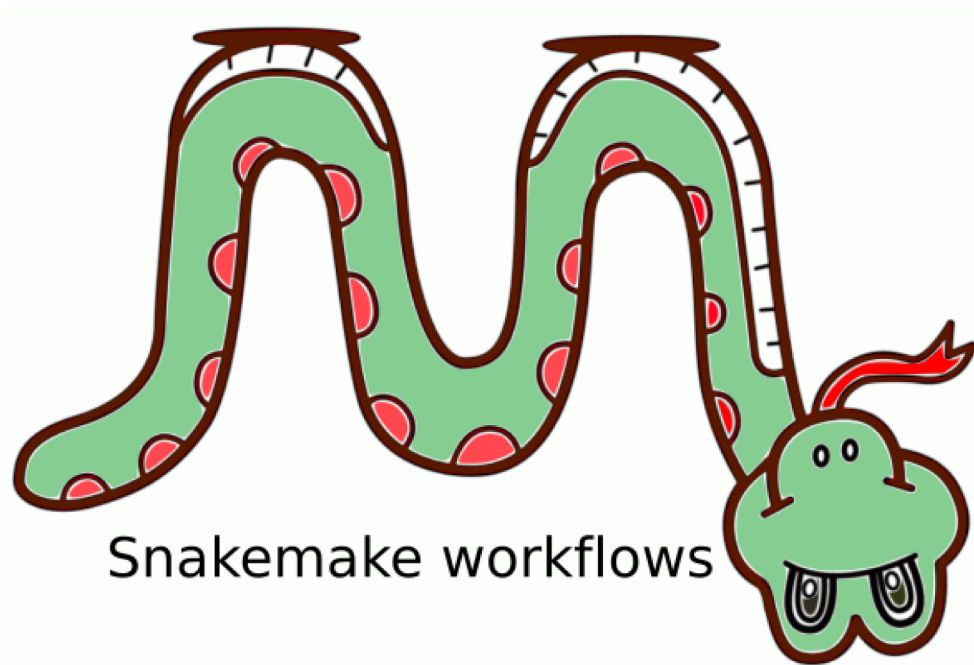
```
## ssh to remote HPC and pull the docker image by singularity
ssh bio1
mkdir singularity-images; cd !$
singularity pull --name rstudio.simg docker://rocker/tidyverse:latest

# 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, /project
# 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-snakemake-ucdavis>
- <https://hackmd.io/jXwbvOyQTqWqpuWwrrpByHQ?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.



Snakemake—a scalable bioinformatics workflow engine

Publication Article in **Bioinformatics**, published October 2012

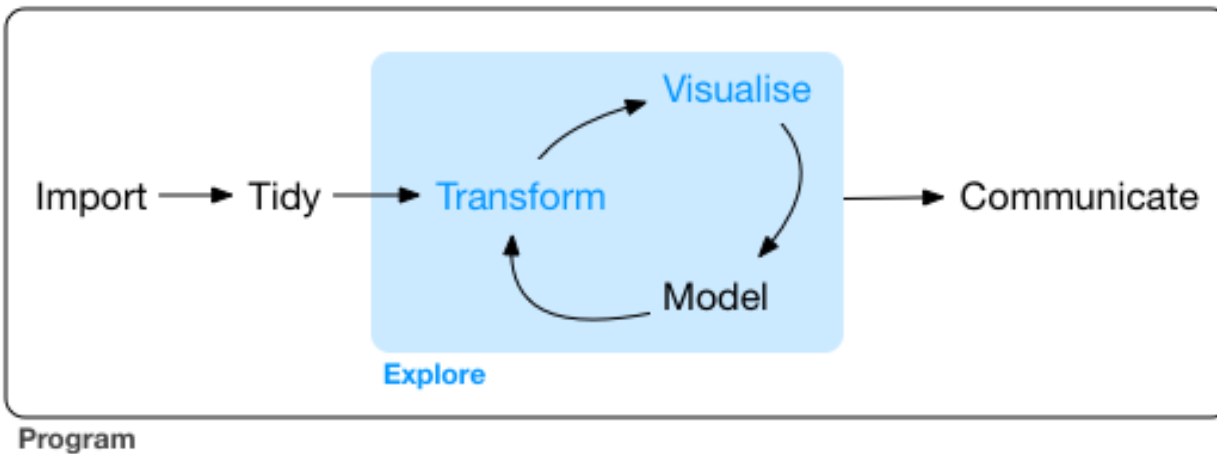
Authors Johannes Köster, Sven Rahmann

[↓ More details](#)



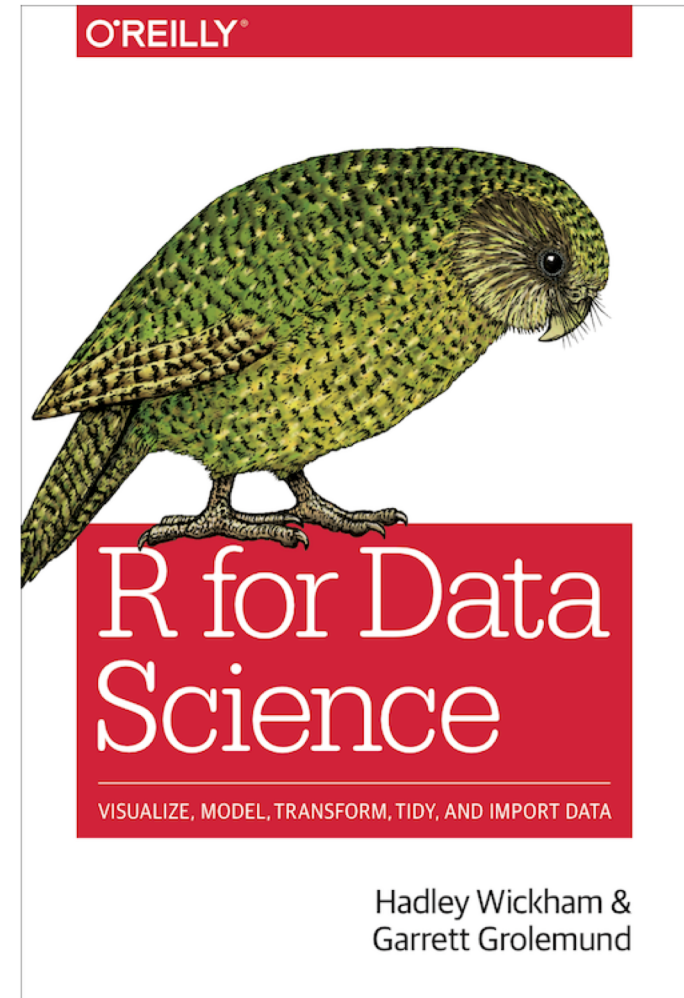
nextflow

Downstream analysis

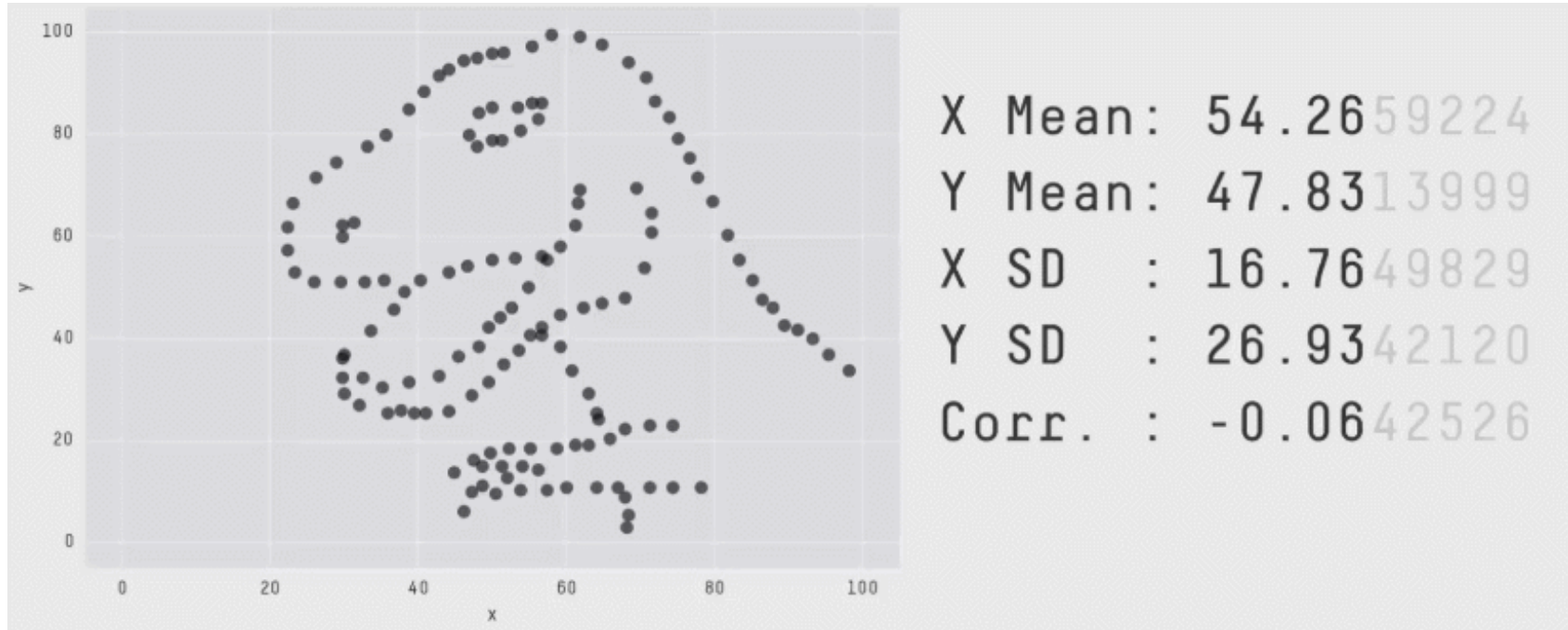


Tidying the data can take 80% of your time
Tidyverse

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



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
```

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
```