

Reproducible genomic data science



Ming ‘Tommy’ Tang

Director of computational Biology at Immunitas

Twitter: tangming2005

<https://divingintogeneticsandgenomics.rbind.io/>

07/09/2022 ISCB2022



Who am I ?



Ming Tang
crazyhottomy

Director of Computational Biology at Immunitas working on single-cell RNAseq. Care about reproducible research and open science

[Edit profile](#)

1.7k followers · 39 following

 Immunitas
 Waltham, MA
 tangming2005@gmail.com
 <http://divingintogeneticsandgenomics.r...>

Achievements

[Overview](#) [Repositories 141](#) [Projects](#) [Packages](#) [Stars 534](#)

crazyhottomy / README.md

Hi there 🤙

- I am a computational biologist working on (single-cell) genomics, epigenomics and transcriptomics.
- I use machine learning approaches to find new drug targets for cancer patients;
- I use google cloud and Terra for large scale data processing;
- 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.

Learn more about me at my [blog](#)

Pinned

 [Chip-seq-analysis](#) Public

Chip-seq analysis notes from Ming Tang

 Python  583  267

 [RNA-seq-analysis](#) Public

RNAseq analysis notes from Ming Tang

 Python  688  262

 [getting-started-with-genomics-tools-and-resources](#) Public

Unix, R and python tools for genomics and data science

 Shell  758  253

 [pyflow-ChIPseq](#) Public

a snakemake pipeline to process ChIP-seq files from GEO or in-house

 Python  89  39

 [scRNaseq-analysis-notes](#) Public

scRNaseq analysis notes from Ming Tang

 373  110

 [scclusteval](#) Public

Single Cell Cluster Evaluation

 R  61  8

Customize your pins

<https://github.com/crazyhottomy>

Reproducibility crisis



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...)

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 Rosic¹, Nicola D. Roberts³, 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.

Why reproducibility is hard?

Why reproducibility is hard?

- 1. no raw data are available.
- 2. scripts/data available upon reasonable request 😊
- 3. lack of method description.
- 4. versions of the tools are different. (e.g. R/python/bioinformatics tools)
- 5. 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.
- I want to make sure my analysis is reproducible because I am discovering drug targets for patients!

How to ensure reproducibility

- Git version control
- Jupyter/R Notebook, documentation
- Containers (docker, singularity, biocontainers <https://biocontainers.pro/>)
- Unit test
- Continuous Integration/development CI/CD (Travis CI, github action)

"FINAL".doc



FINAL.doc!



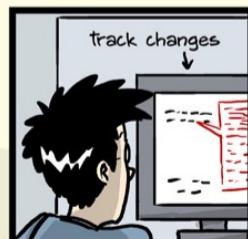
FINAL_rev.2.doc



↑
FINAL_rev.6.COMMENTS.doc



↑
FINAL_rev.8.comments5.
CORRECTIONS.doc



↑
FINAL_rev.18.comments7.
corrections9.MORE.30.doc



↑
FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL????.doc



JORGE CHAM © 2012

Version control

- Git
- Github
- Gitlab



Jupyter Notebook



[JUPYTER](#) [FAQ](#) </>

[notebook](#) / [docs](#) / [source](#) / [examples](#) / [Notebook](#)

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, therefore 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` runs the current cell and enters command mode.

R notebook/markdown

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

The screenshot shows the RStudio Source Editor interface. The top bar displays the title "RStudio Source Editor" and the file name "nb-demo.Rmd". Below the title bar is a toolbar with various icons for file operations, search, and preview. The main area contains R code in a syntax-highlighted editor. The code includes two code chunks (indicated by triple backticks) and one regular text chunk. The first code chunk uses the `summary` function on the `iris` dataset, and its output is a table showing descriptive statistics for Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species. The second code chunk loads the `ggplot2` library and creates a scatter plot of Sepal.Length versus Petal.Length, where points are colored by Species and sized by Petal.Width. The plot shows three distinct clusters corresponding to the species: setosa (light blue), versicolor (medium blue), and virginica (dark blue).

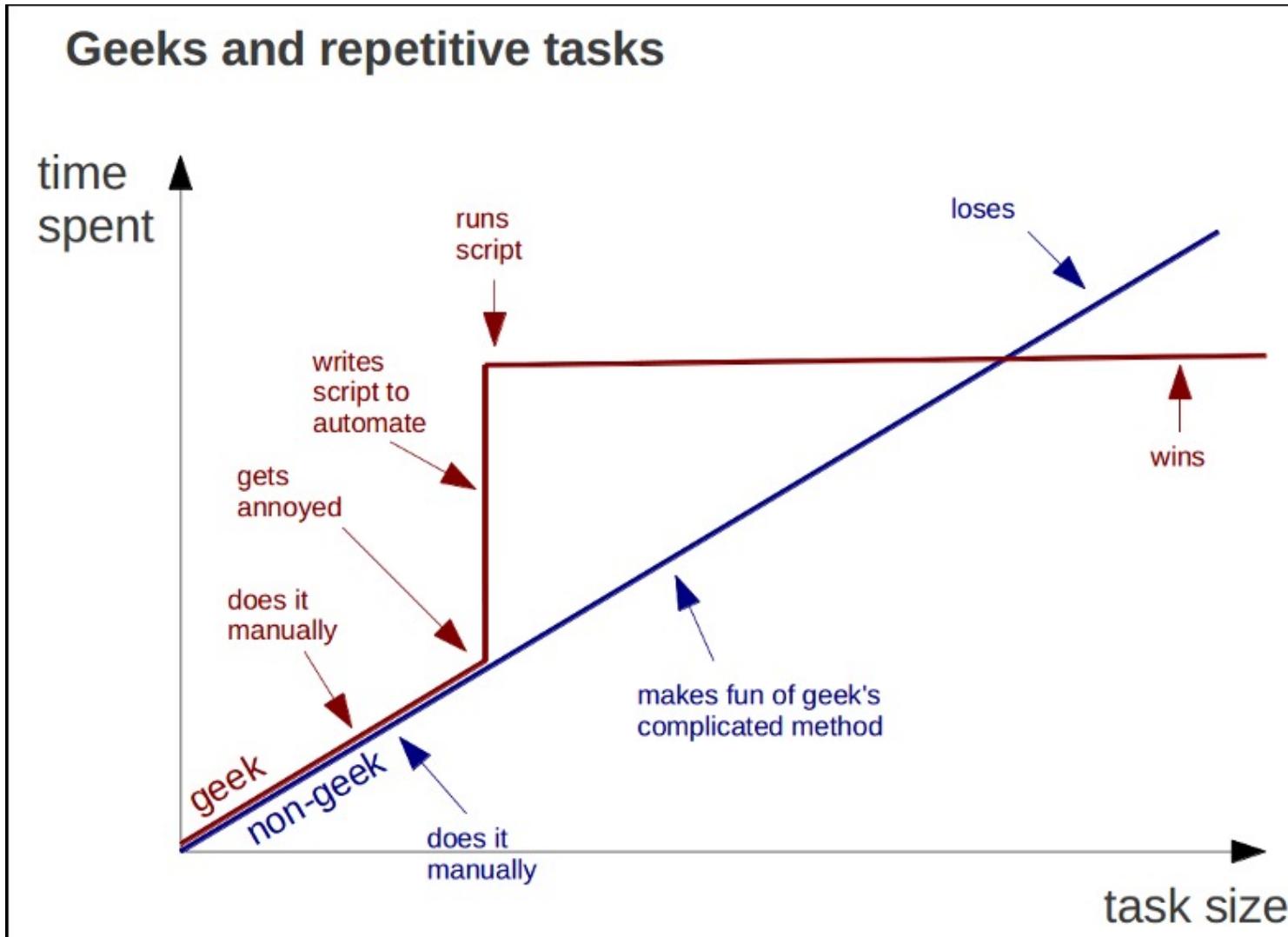
```
9
10 ````{r}
11 summary(iris)
12 ````

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   :4.358 Mean   :1.588
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

13
14 ````{r}
15 library(ggplot2)
16 qplot(Sepal.Length, Petal.Length, data = iris, color = Species, size =
Petal.Width)
17 ````

  
```

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

- The best documentation is automation
- Write scripts for everything unless it is not possible. (manual editing, document, document, document!)
- Markdown, MKdocs <https://www.mkdocs.org/>

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 GNU make
- 3. 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.



Snakemake—a scalable bioinformatics workflow engine

Publication Article in *Bioinformatics*, published October 2012

Authors Johannes Köster, Sven Rahmann

[More details](#)



nextflow

docker



- 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.
- Think it as a virtual machine!
- This just happened between me and my colleagues who used a different version of R packages!

<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>

conda and biocoda

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

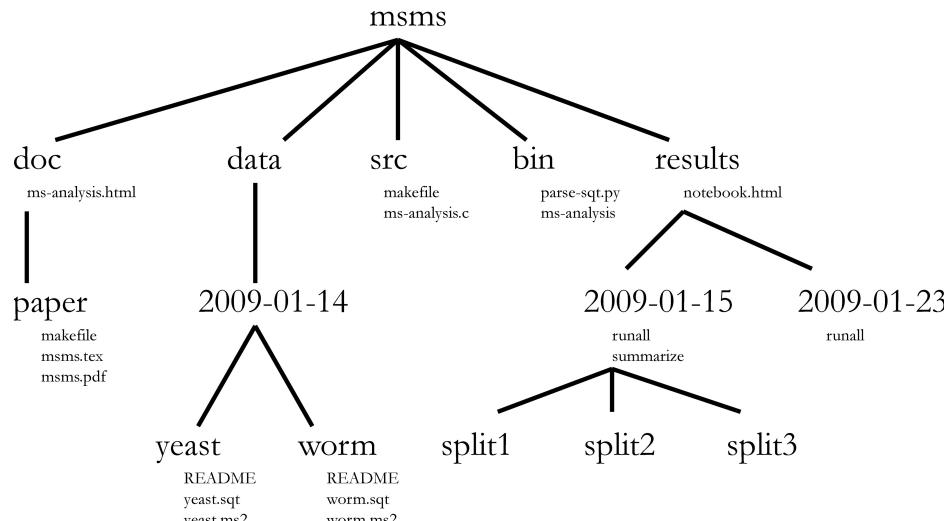
Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A. Chapman, Jillian Rowe, Christopher H. Tomkins-Tinch, Renan Valieris & Johannes Köster ✉ The Bioconda Team

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

 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

Workflowr for fully reproducible analysis



Belinda Phipson
@BelindaPhipson

Check out this incredibly impressive workflowr analysis website showcasing @JovMaksimovic single cell analysis of paediatric lower airway. A lot of time and effort to ensure the analysis is reproducible.
oshlacklab.com/paed-cf-cite-s...



biorxiv.org
Multimodal single cell analysis of the paediatric lower airwa...
Respiratory disease is a major cause of morbidity and mortality in children worldwide. Many childhood respiratory...

1:10 AM · Jun 24, 2022 · Twitter Web App

16 Retweets 2 Quote Tweets 60 Likes



paed-cf-cite-seq Home About License

Abstract

Authors
Analysis Overview
Licenses
Citations
Version Information

Multimodal single cell analysis of the paediatric lower airway reveals novel immune cell phenotypes in early life health and disease

Jovana Maksimovic
2022-06-20



This site presents the code and results of the analyses described in the pre-print: "Multimodal single cell analysis of the paediatric lower airway reveals novel immune cell phenotypes in early life health and disease".

All the code and results of this analysis are available from GitHub at <https://github.com/Oshlack/paed-cf-cite-seq>. To reproduce the complete analysis follow the instructions on the [getting started](#) page. The raw single cell RNA-seq and CITE-seq count data generated for this study can be downloaded as RDS files from DOI [10.5281/zenodo.6651465](https://doi.org/10.5281/zenodo.6651465).

Follow the links below to view the different parts of the analysis.

Thursday, August 13, 2015

2 cents on coding from a bioinformatics beginner

One needs to be aware that:

1. **Computers make mistakes.** They can give you non-sense results and exit without error, so make extensive tests before running your code.
2. **Share your codes.** Even your codes are correct, you need to share them so that other people can look at them and may improve them.
3. **Make your codes reusable.** Do not hard code your scripts. If it takes a file path as input, make it as an argument in your scripts.
4. **Modulate your scripts.** Data could come in different stage of formats. Take ChIP-sequencing data analysis as an example, if you have a script that starts processing the data from fastq to the final peaks. You may want to modulate your scripts to two modules: one for mapping fastq to bam, and the other for bam to peaks. **Modulate your scripts** so that one can use your script when the data come in a bam format.
5. **Heavily comment your scripts.** It will not only make other people to understand your codes better, but also help the future you to understand what you did.
6. **You need to make your analysis reproducible.** Each step of your analysis should be documented in a markdown file. I say every step, yes, every command that you strike in the terminal getting the intermediate files need to be taken down. Moreover, how, when and where did you download the data need to be documented. This will save the future you! Many experienced programmers overlook this point.

Acknowledgments

Verhaak Lab
Samir Amin

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