

# a software developer and a data scientist *in an open science world*

lennart martens

*[lennart.martens@ugent.be](mailto:lennart.martens@ugent.be)*

*computational omics and systems biology group*

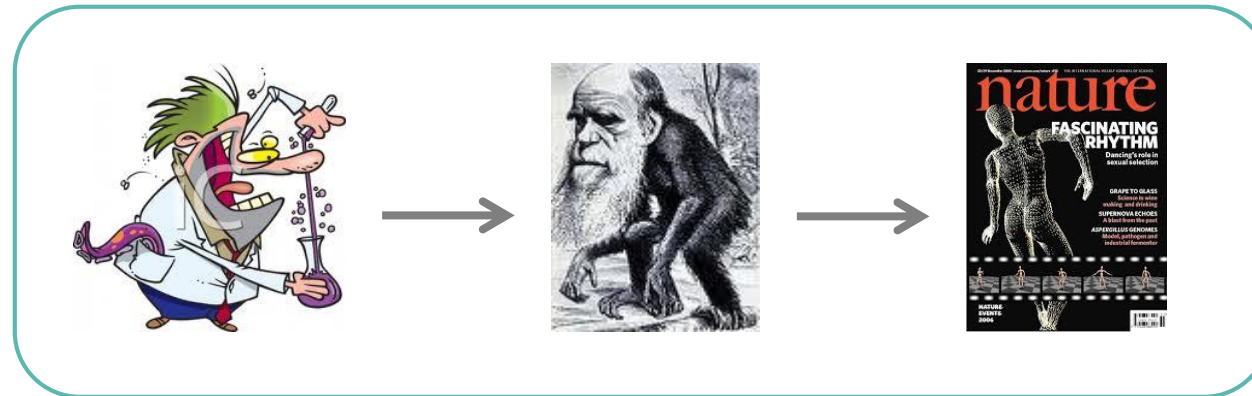
*Ghent University and VIB, Ghent, Belgium*



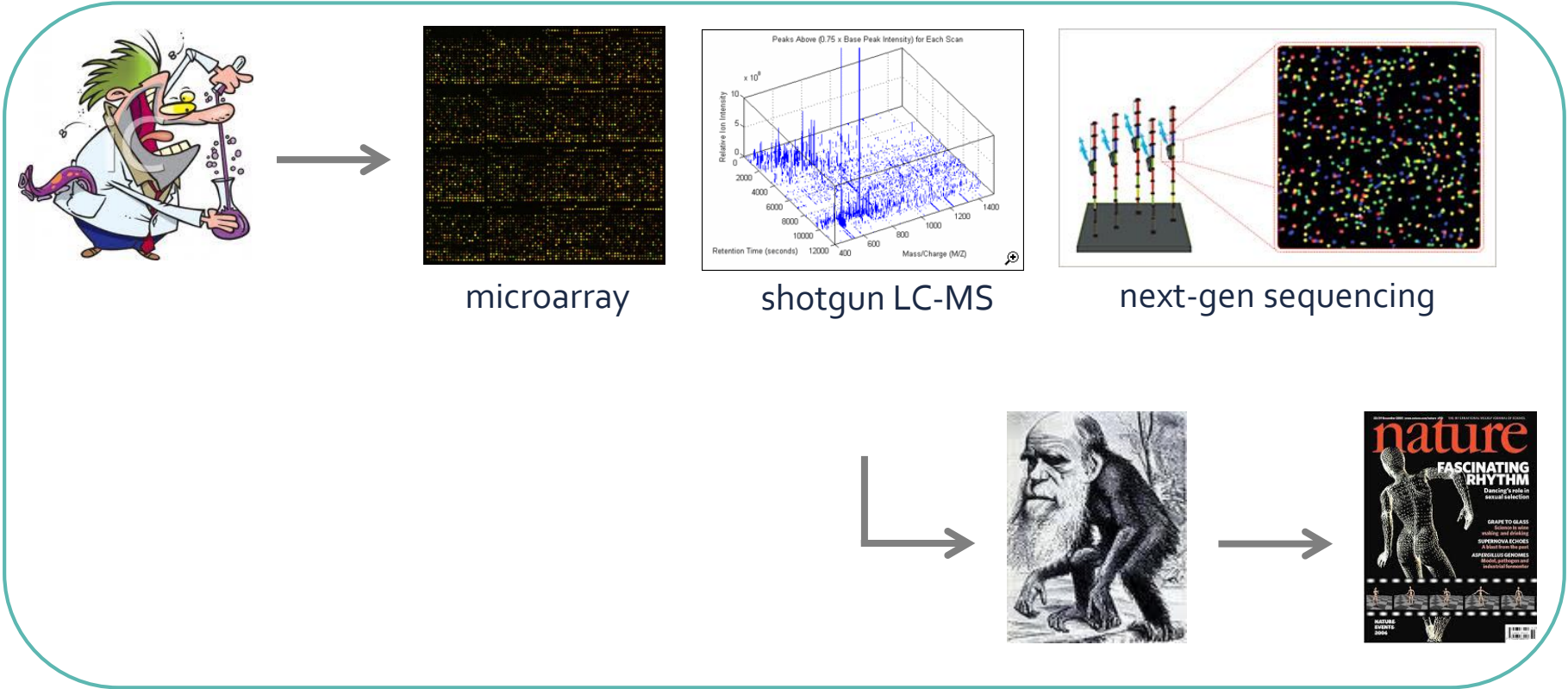
CC BY-SA 4.0



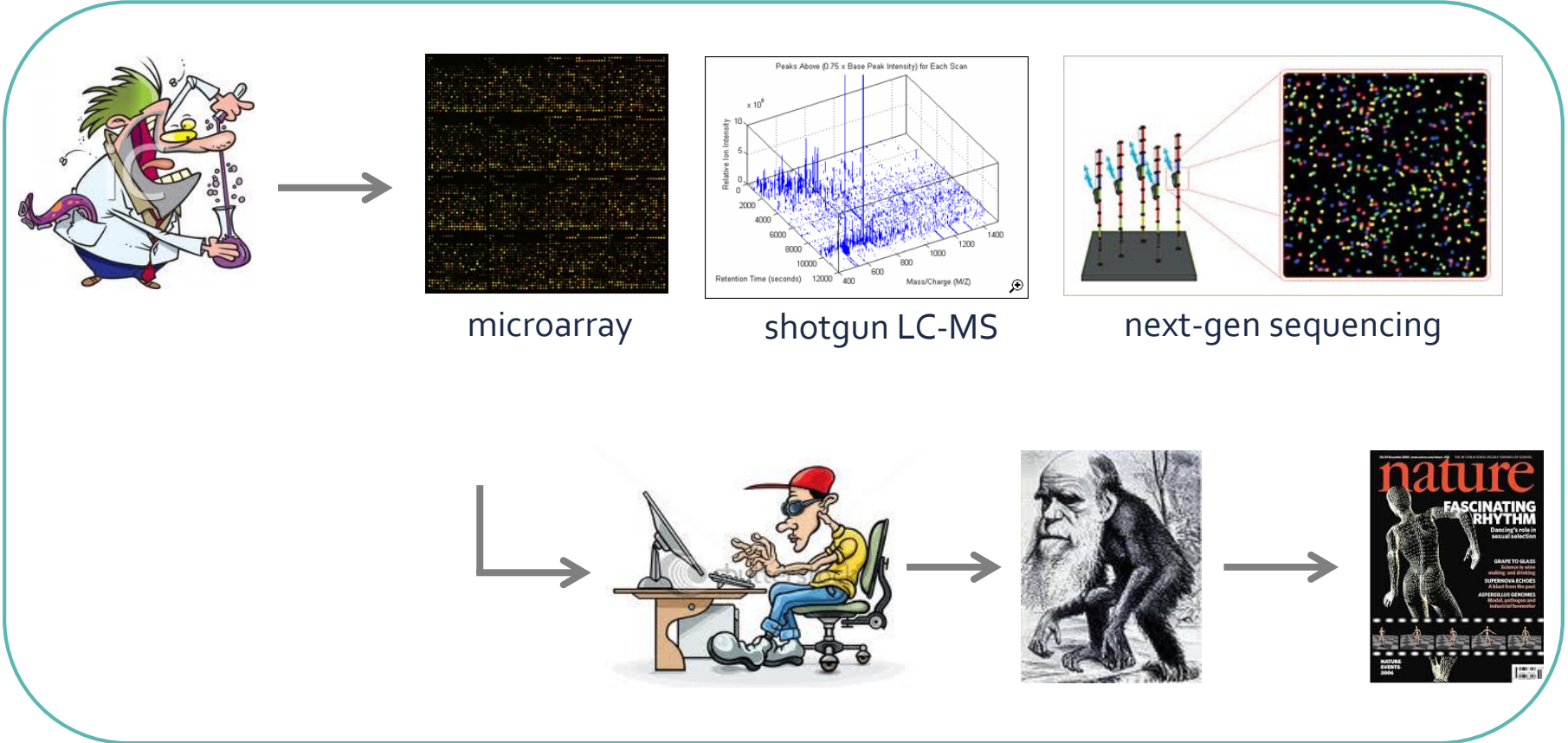
# Not too long ago, life sciences researchers managed very well without computers



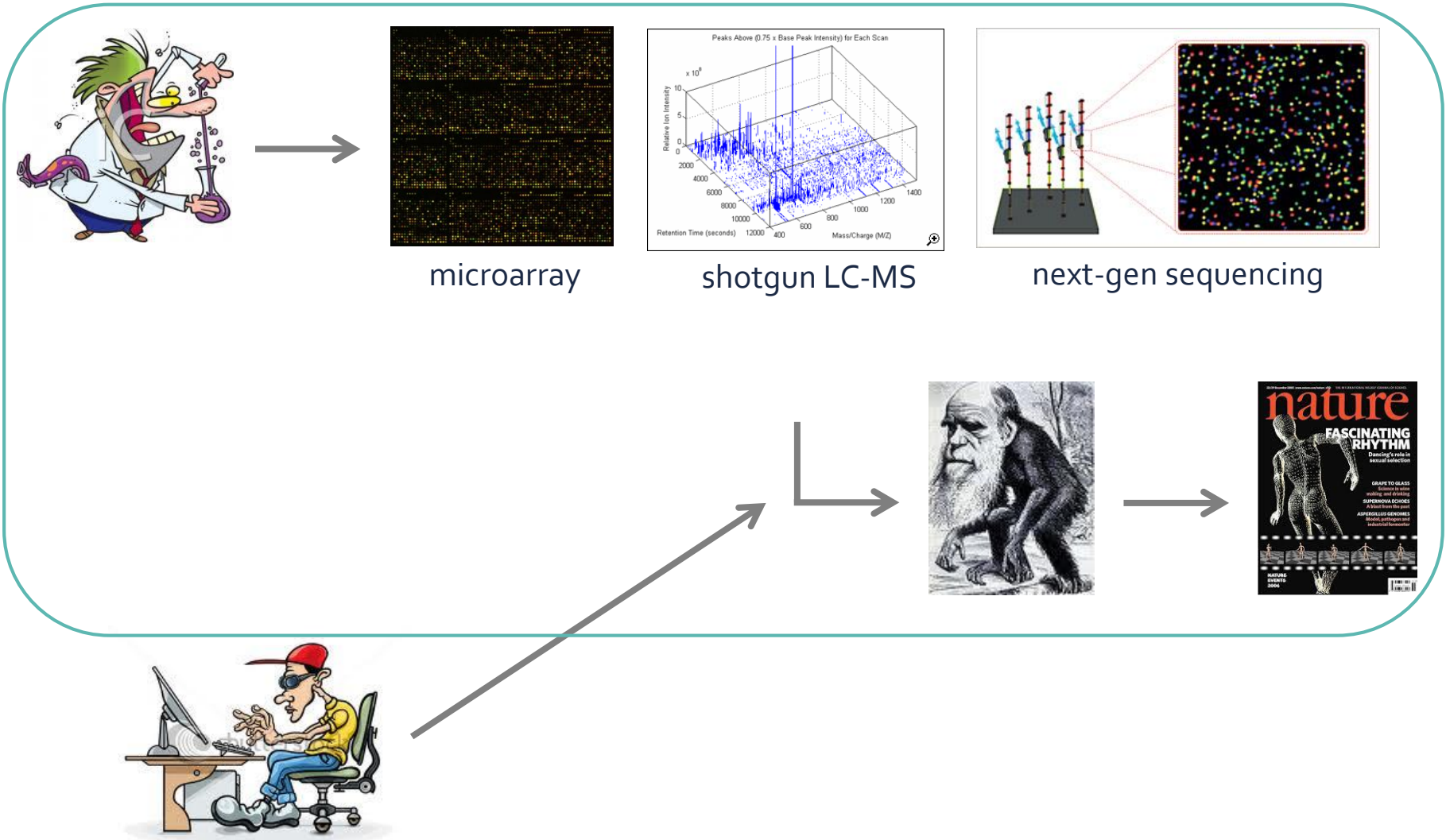
# But then these researchers embraced technology, and its expensive instruments



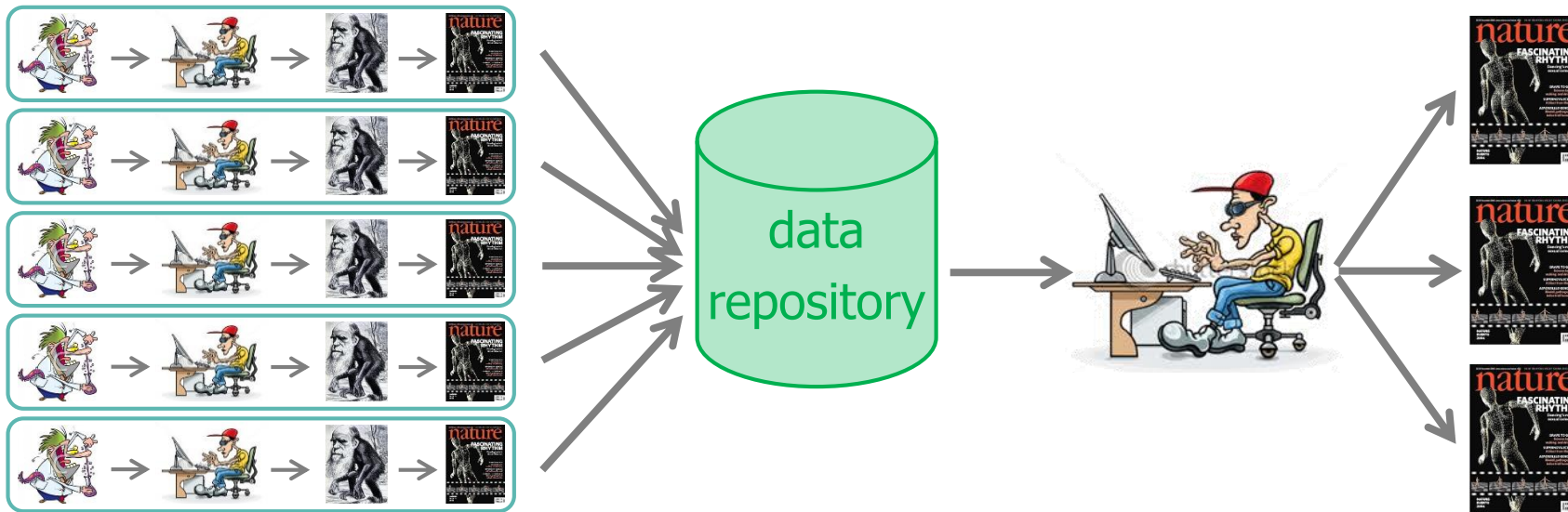
# And the embedded bioinformatician arrived in the life sciences research team



# At the same time, the Researcher-Developer started to gain prominence as well

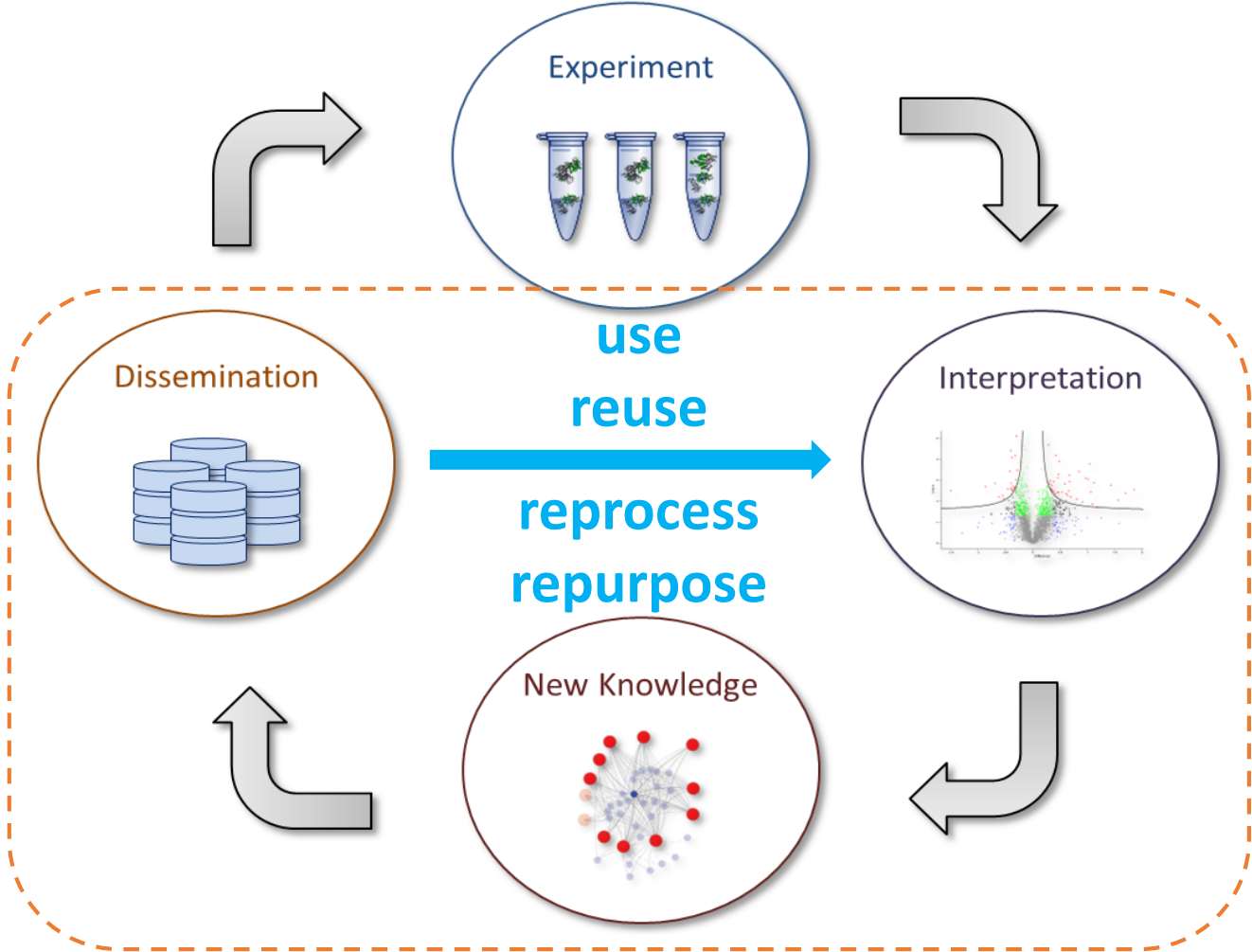


# And now we have entire research groups simply maximising their keyboard time!



# An open data exchange ecosystem allows for productive (and completely novel!) data uses

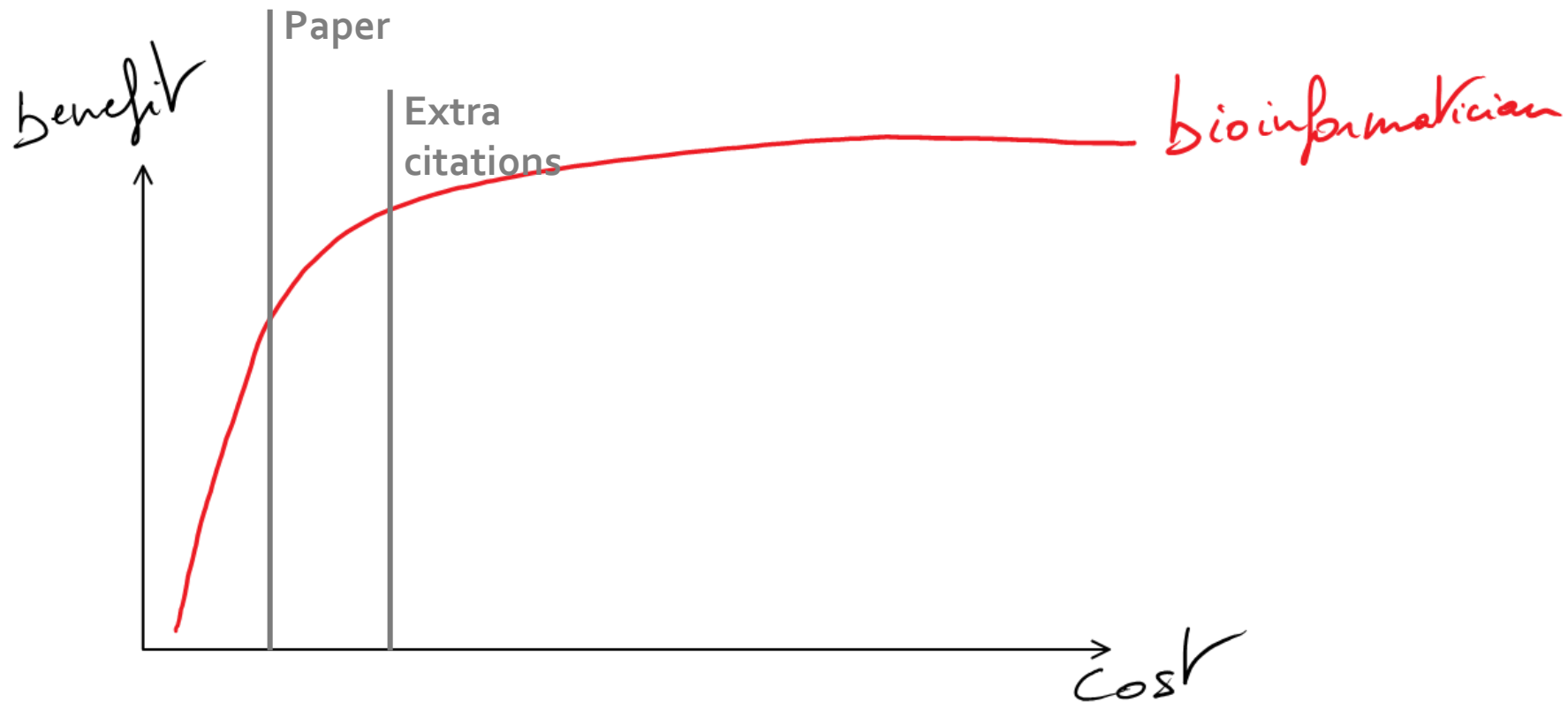
## Open Data Exchange Ecosystem



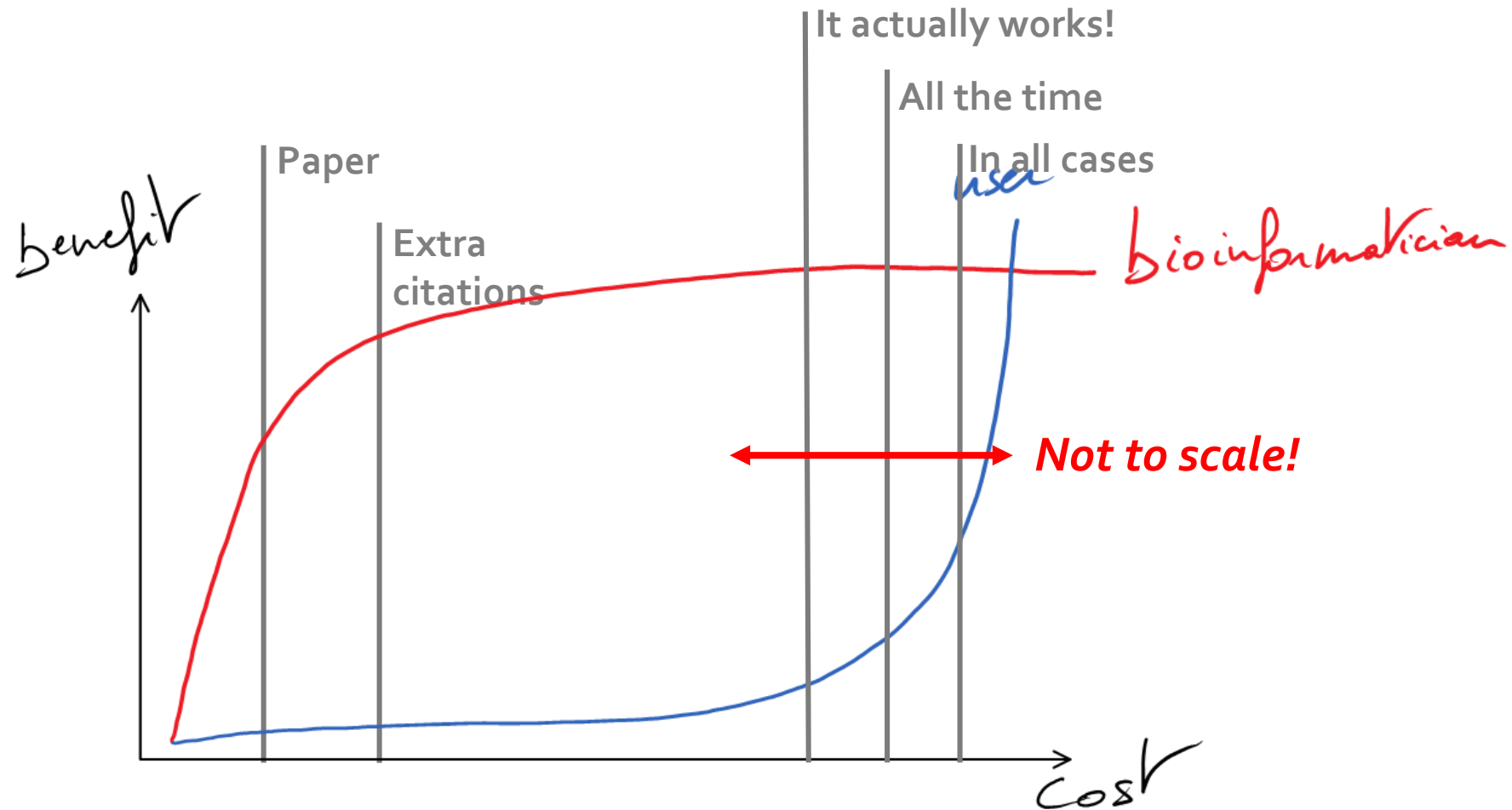
Adapted from: Vaudel, Proteomics, 2016



# Developing (academic) software presents a particular diminishing returns curve



# Which it is good to keep in mind when you start



Making things that actually work can be harder than it looks,  
but it should not discourage you!

Target: peppermint icicles



Result: peppermint ... eh ... erm...



Many academic informatics approaches, even published, may not be meant for (all) users

## Computational Proteomics: Managing Expectations

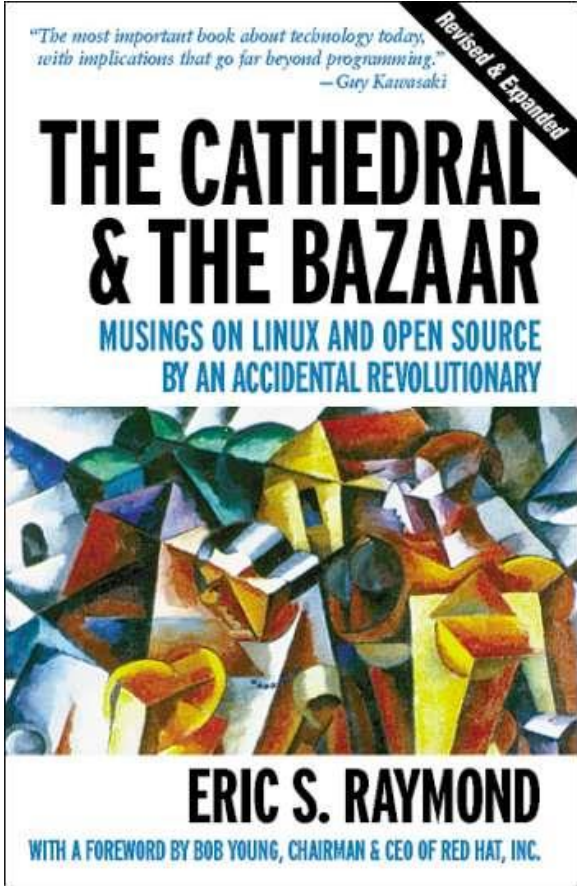


# Three manuscript types with different requirements in the Journal of Proteome Research

**Table 1. Computational Manuscript Types and Their Requirements**

	brief communication	research article	application note
substantial advancement	+	+	
potential for reuse		+	+
general limitations	+	+	+
system limitations		+	+
end-user documentation	+	+	+
developer documentation		+	+
sample data		+	+
benchmark data set		+	+
availability		+	+
license information		+	+
system requirements		+	+

The open source paradigm is old and venerable, and certainly not only linked to science



# The changing point of view of Microsoft provides an interesting angle on FOSS



2001, Steve Ballmer, CEO of Microsoft:  
“Linux is a cancer that attaches itself in an intellectual property sense to everything it touches.”

2018, Satya Nadella, CEO of Microsoft:  
“We are all in on open source.”



Open source code allows others to re-use, to correct, and to improve your software

Few, if any, developers build all their code from scratch these days

By building on code of others, we can focus on our work, rather than on allowing this work to be done

Your benefits translate to other (younger) researchers, so we're paying forward

If your work is useful and successful, others might start to contribute and enhance your work!



Open source software should ideally be hosted by a (reliable) third party

Free and commercial hosters exist (often these are the same entities) for your software, for instance GitHub, BitBucket, and SourceForge

These hosters typically offer useful features such as version control and issue trackers

More and more, social features are also a key part of the framework of these hosters

It is conceivable that your CV will list the number of pull requests from GitHub, from instance

# Open source code should be hosted on a third-party platform, like GitHub, BitBucket, or similar

The screenshot shows the GitHub profile page for the 'compomics' organization. The profile name is 'Computational Omics and Systems Biology Group', with a bio stating they specialize in high-throughput Omics data. The page features a grid of pinned repositories: 'ThermoRawFileParser' (C#, 150 stars), 'DeepLC' (Python, 43 stars), 'ms2rescore' (Python, 32 stars), 'compomics-utilities' (Java, 26 stars), 'peptide-shaker' (Java, 41 stars), and 'searchgui' (Java, 35 stars). Below the pinned repositories is a 'Repositories' section with a search bar and filters for Type, Language, and Sort. The first repository listed is 'ms2rescore', followed by 'DeepLC'. The right sidebar includes a 'Follow' button, a 'View as: Public' dropdown, a 'Discussions' section, a 'People' section with a grid of member avatars, and a 'Top languages' section showing Java, Python, HTML, Jupyter Notebook, and JavaScript.

<https://github.com/compomics>



CC BY-SA 4.0

# When it comes to the analysis of your data, your paper contains the advertisement...

nature.com | Publications A-Z index | Browse by subject | Access provided to University Library Gent by Rozier 9 | Login | Register | Cart

nature genetics

Search  Go [Advanced search](#)

Home | Current issue | Comment | Research | Archive | Authors & referees | About the journal

home > archive > issue > analysis > full text

NATURE GENETICS | ANALYSIS

日本語要約

## Multi-tiered genomic analysis of head and neck cancer ties *TP53* mutation to 3p loss

Andrew M Gross, Ryan K Orosco, John P Shen, Ann Marie Egloff, Hannah Carter, Matan Hofree, Michel Choueiri, Charles S Coffey, Scott M Lippman, D Neil Hayes, Ezra E Cohen, Jennifer R Grandis, Quyen T Nguyen & Trey Ideker

[Affiliations](#) | [Contributions](#) | [Corresponding author](#)

Nature Genetics 46, 939–943 (2014) | doi:10.1038/ng.3051  
Received 20 March 2014 | Accepted 10 July 2014 | Published online 03 August 2014


[PDF](#) [Citation](#) [Rights & permissions](#) [Article metrics](#)

### Abstract

[Abstract](#) • [Introduction](#) • [Results](#) • [Discussion](#) • [Methods](#) • [References](#) • [Acknowledgments](#) • [Author information](#) • [Supplementary information](#)

Head and neck squamous cell carcinoma (HNSCC) is characterized by aggressive behavior with a propensity for metastasis and recurrence. Here we report a comprehensive analysis of the molecular and clinical features of HNSCC that govern patient survival. We find that *TP53* mutation is frequently accompanied by loss of chromosome 3p and that the combination of these events is associated with a surprising decrease in survival time (1.9 years versus >5 years for *TP53* mutation alone). The *TP53*-3p interaction is specific to chromosome 3p and validates in HNSCC and pan-cancer cohorts. In human papillomavirus (HPV)-positive tumors, in which HPV inactivates *TP53*, 3p deletion is also common and is associated with poor outcomes. The *TP53*-3p event is modified by mir-548k expression, which decreases survival further, and is mutually exclusive with mutations affecting RAS signaling. Together, the identified markers underscore the molecular heterogeneity of HNSCC and enable a new multi-tiered classification of this disease.

### Editors' pick



Focus on TCGA Pan-Cancer Analysis >

### Science jobs

Science events

**naturejobs.com**

Manuscript Assistant / Publishing Assistant - Talent Pool 2017  
Springer Nature

Postdoc position at Johns Hopkins University  
Johns Hopkins University

Corporate Communications Manager, Springer Nature, Japan  
Springer Nature

[Post a job](#) | [More science jobs](#)

### Discover more

Most read

Small-RNA asymmetry is directly driven by mammalian Argonautes  
Nature Structural & Molecular Biology | 22 Jun 2015

Emerging biomarkers in head and neck cancer in the era of genomics  
Nature Reviews Clinical Oncology | 18 Nov 2014

# ... but the code on GitHub represents the actual research performed

theandygross / TCGA

No description, website, or topics provided.

212 commits | 1 branch | 2 releases

File	Description
Analysis_Notebooks	Pre-package split.
Extra_Data	Move imports to separate notebook.
src	Pre-package split.
.gitignore	fix .gitignore
README.md	split README

README.md

### #Software Overview

This repository contains instructions for reproduction and extension of [Multi-tiered genomic analysis of head ties TP53 mutation to 3p loss](#) by Gross et al. In general code for data-processing and computation is enclosed python modules, while high level analysis was recorded in IPython Notebooks. The analysis for this project was linear and has thus been split into a number of notebooks as described in [Analysis Notebooks](#), but results should be replicated by running these notebooks.

## HNSCC Cohort

Here we include a general analysis of HPV within the first round of TCGA patients (which eventually became the discovery cohort after filters). Most of this analysis was not used in the final paper, but it can be seen that there are very clear global difference: HPV+ and HPV- patients which would confound analysis of these patients as a combined cohort.

### Import Data and Packages

For full list of data and packages imported see the [Imports](#) notebook.

```
In [1]: import NotebookImport
from Imports import *
```

importing IPython notebook from Imports.ipynb  
Populating the interactive namespace from numpy and matplotlib  
changing to source directory  
populating namespace with data

For the majority of the analysis we use the RNA and mRNA datasets filtered down to HPV- patients. Here we use the full datasets, we need to reload them on top of the others (which are loaded by default in the Imports notebook).

```
In [2]: rna = cancer.load_data('mRNASeq')
mirna = cancer.load_data('miRNASeq')
pppa = cancer.load_data('RPPA')
```

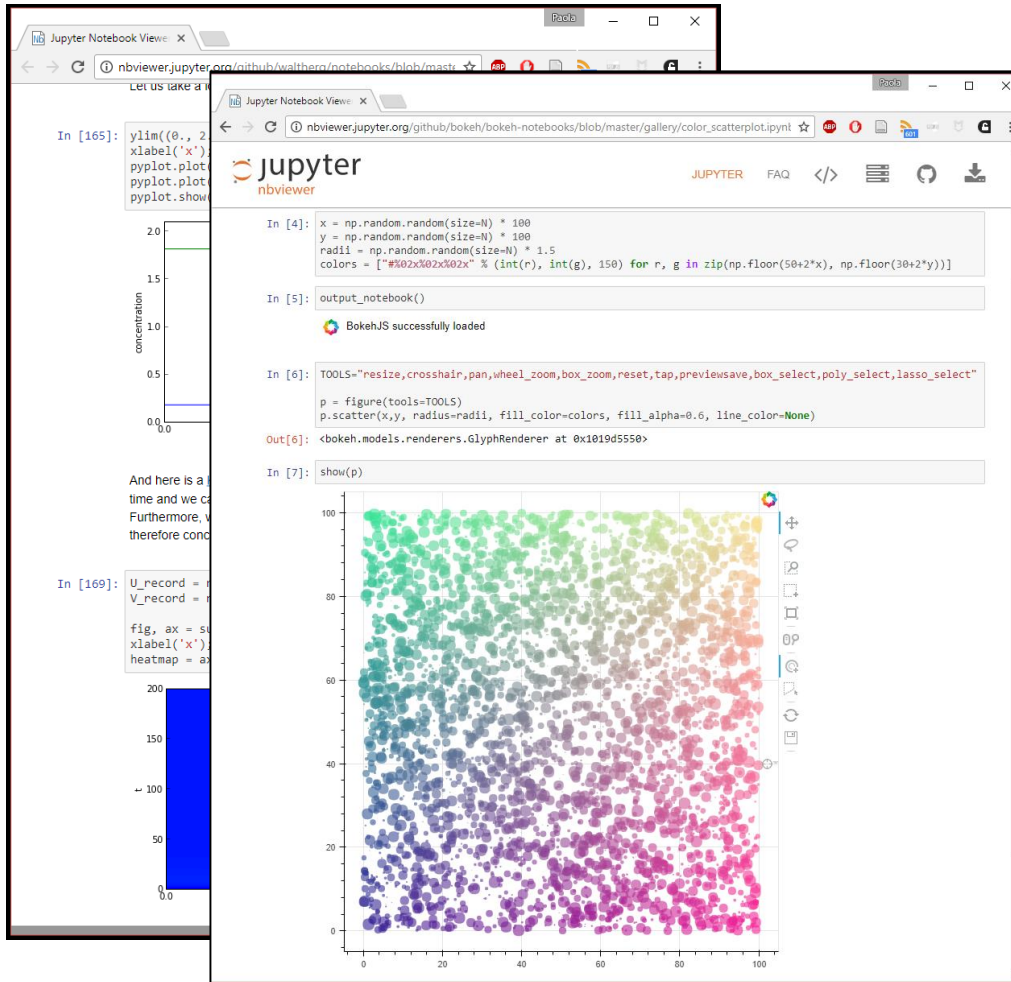
### HPV Clinical Correlates

```
In [3]: hpv = hpv.map({True:'HPV+', False:'HPV-'})
```

```
In [4]: fig, ax = subplots(figsize=(5,3))
draw_survival_curve(hpv, surv, ax=ax)
ax.legend(title=False, frameon=False, loc='lower left')
prettify_ax(ax)
fig.tight_layout()
fig.savefig(FIGDIR + 'hpv_sup_a.pdf', transparent=True)
```

Years	HPV- Survival	HPV+ Survival
0	1.0	1.0
1	0.8	0.95
2	0.6	0.9
3	0.5	0.85
4	0.45	0.75
5	0.4	0.5

# Interactive notebooks enable development, code sharing, and reporting all in one place



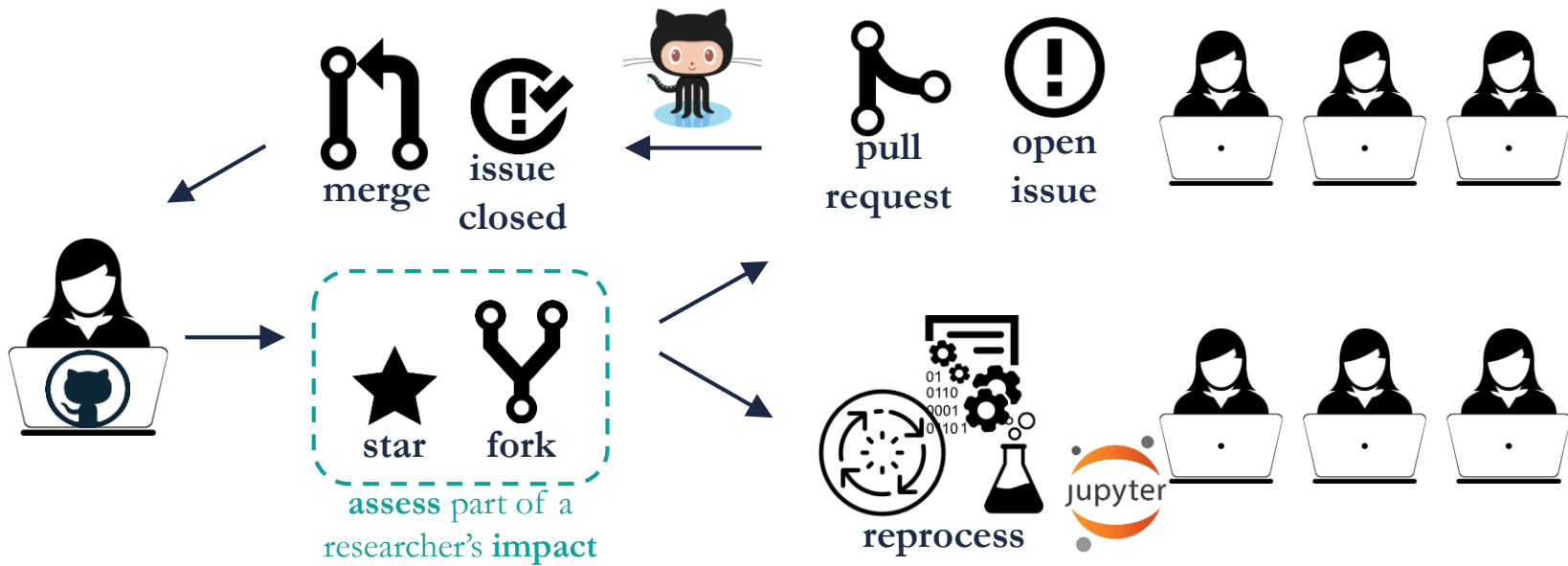
a browser-based and interactive notebook with support for code, rich text, mathematical expressions, inline plots and other rich media

an ideal platform to support **open** and **reproducible** research

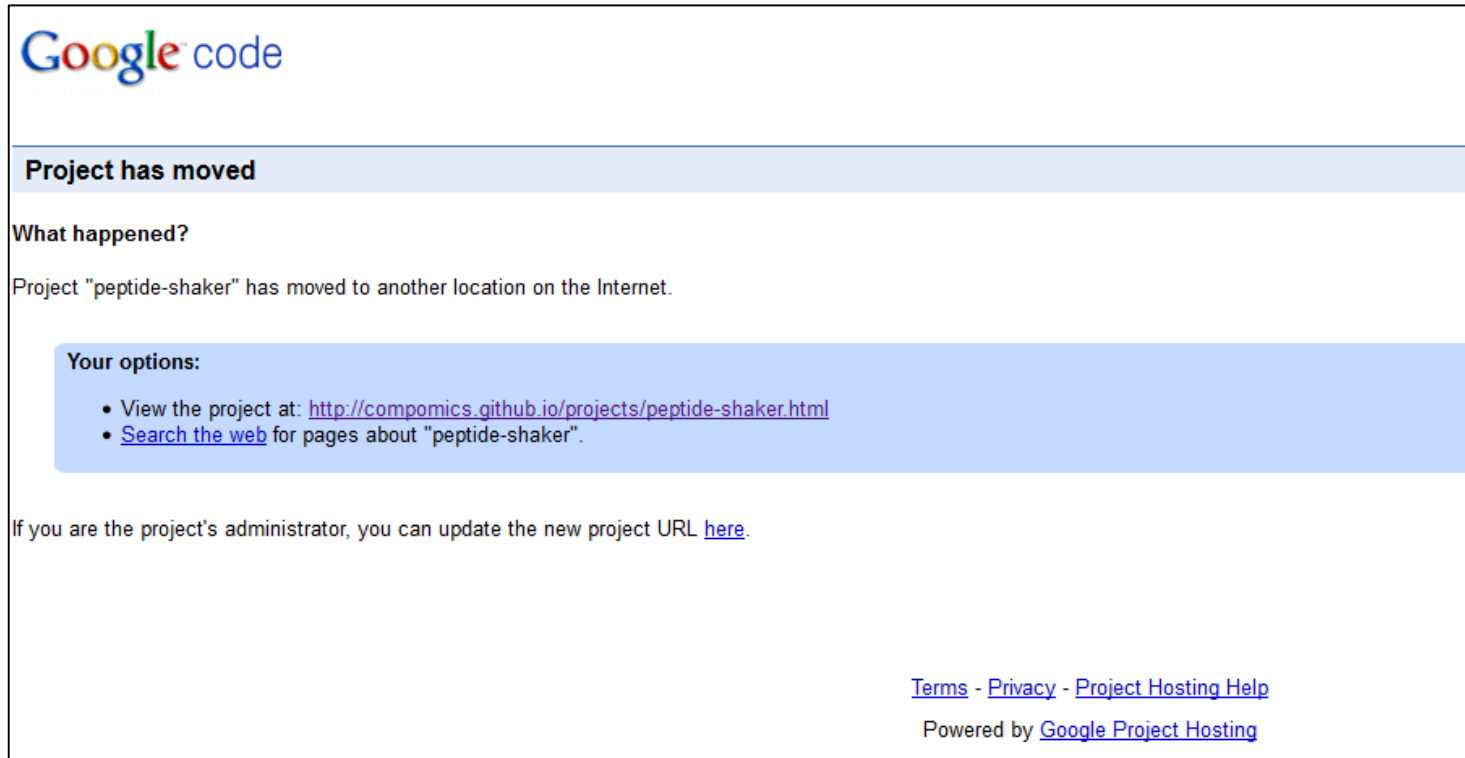


Technically, a Jupyter notebook could easily be a publication!

# Open code allows collaboration as well as reproduction



# As a responsible caretaker of your stuff, you will sometimes need to take action too



The screenshot shows a notification page from Google Code. At the top left is the 'Google code' logo. Below it is a light blue header bar with the text 'Project has moved'. Underneath, the text reads 'What happened?' followed by 'Project "peptide-shaker" has moved to another location on the Internet.' A light blue box contains the heading 'Your options:' and two bullet points: 'View the project at: <http://compomics.github.io/projects/peptide-shaker.html>' and 'Search the web for pages about "peptide-shaker"'. Below this box, it says 'If you are the project's administrator, you can update the new project URL [here](#).' At the bottom right, there are links for 'Terms - Privacy - Project Hosting Help' and 'Powered by [Google Project Hosting](#)'.

# Maintaining what you do is not trivial, not cheap, and may even be considered unwise by senior PIs

Google code

**Project has moved**

What happened?

Project "peptide-shaker" has moved to another location on the Internet.

**Your options:**

- View the project at: <http://compomics.github.io/projects/peptide-shaker.html>
- [Search the web](#) for pages about "peptide-shaker".

If you are the project's administrator, you can update the new project URL [here](#).

[Terms - Privacy](#)

Powered by

Compomics

Home Compomics Github


## PeptideShaker

- Introduction
- Read Me
- Troubleshooting
- Bioinformatics for Proteomics Tutorial

**PeptideShaker Publication:**

\* Vaudel et al. Nature Biotechnol. 2015 Jan;33(1):22-24.  
\* If you use PeptideShaker as part of a publication, please include this reference.

[Download PeptideShaker](#) v1.1.3 - All platforms [ReleaseNotes](#)



LINKS

- Project Home
- Source
- Issues

WIKI

- PeptideShakerCLI
- ReleaseNotes

PROJECTS

- colims
- compomics-utilities
- denovogui
- fragmentation-analyzer
- iceLogo
- icelogoserver
- jsparklines
- jtraml
- mascotdatfile
- ms-lims
- ols-dialog
- omssa-parser
- pepsHell
- ▶ peptide-shaker
- pladipus
- pride-asa-pipeline
- reporter
- scoreburst



# Some best practices for (open source) software development

Document your code, document your APIs

Use unit testing to ensure correct code functioning

Use a versioning system to keep track of changes

Use an issue tracker for your development

Where possible, adopt co-development

Pro tip: a lot of commercial software is free if you agree to only use it for open source development

# My institute (tech transfer) does not allow me to make my code open source

Usually, this is based on perceived potential for commercial exploitation

Talking to the responsible person often resolves issues for software without commercial prospects

Can also be resolved by adopting a suitable licensing policy (but this is hard to do retroactively!)

And remember: software itself does not make money that often, but service and support does

# I wish to commercialize my software

Adopt a licensing policy that makes your source code open and free to academics, but incompatible with commercial exploitation, and provide a separate license for commercial use

Build a model based on service and support rather than on the tool itself (essentially the RedHat model)

Use an open source base, and add value in the commercial suite (eg., integration, ease of use)

I don't want anyone else to tell me  
what I may have done wrong

Get out of science quick! Scientists will constantly tell you what they think  
you do wrong, no code needed

There's nothing wrong with people helping you to correct your stuff – there  
will *always* be mistakes

Often, people who tell you about your errors based on your code will  
actually suggest or provide a solution!

# I don't want anyone to look at my code; it is horrible!

Try to write clean code, regardless of whether it is open; your future self will thank you profusely!

Moreover, bad code is altogether not very useful; it tends to be unmaintainable and buggy

Coding best practices are actually easy, and can be learned online for your language of choice

Having your code out in the open is a great way to motivate yourself to write better code, the primary beneficiary of which will be you!

I want to maximize my CV by requiring people to go through me to use my software

While this may sound like an attractive prospect, it is unlikely to land you a great career

Committees (for funding, tenure, important stuff) look increasingly at your own achievements and less at bean counting (such as number of co-authorships)

A better model here would be to start a commercial entity and offer your expertise and tools as a service

Your collaborators might have a hard time publishing their results, especially if open practices are competing

My script is so useless, simple, and unimportant that opening it makes no sense

Fair enough, but why not make it available as supplementary information then?

It costs nothing, it is unlikely to harm you, and in the worst case scenario, reviewers ask you to clean it up

Others are going to scoop me when they steal my open source code

Technically, open source code cannot really be stolen 😊

By using a third-party hoster with versioning, you can always claim and prove priority when it comes to this

In practice, these situations are few and far between

An effective measure is peer pressure: consistently name and shame people who actually do this



# 'Researcher-Developer' is a real job, and should be treated as such

Building usable tools is sufficiently complex that it requires a separate job title, and separate specialization

this however, does NOT mean that you can treat tools as black boxes; make sure you know what happened to your data

commoditization is an ongoing process in research software; learn to take advantage of it (*but see next slide for caveats*)

typical academic informatics analyses are not an afterthought in a study; instead these have become a substantial part of a project

fundors increasingly require data analysis and management plans; this is specifically meant so that you have your bioinfo thought out, planned and *funded* in your project

for full clarity: Researcher-Developers are not IT helpdesks

# Commoditization of academic informatics tools is typically haphazard at best

Academic informatics (and tool development) is too often an afterthought in the project, which shows in the results

tool development is actually considered as irrelevant by many experimentalists because their focus is on getting the data

in rare (but highly unfortunate!) cases, software tools can be considered competitive and are not shared

development of user-usable solutions is currently heavily counter-incentivized (*as shown previously*)

companies can provide good solutions, but cutting-edge approaches tend to appear later in commercial solutions

many groups are constantly re-inventing the wheel, or perhaps better put: the rubber, vulcanization, and the inner tube

# A sociologist's take on our efforts towards (orthogonal) data reuse

"This desire to reactivate data is widespread, and Klie et al. are not alone in wanting to show that 'far from being places where data goes to die' (Klie et al., 2007: 190), **such data collections can be mined for valuable information that could not be obtained in any other way.**"

"In attempting to **reactivate sedimented data** in order to enable its re-use, their first step was ..."

"... they are experiments in seeing, in furnishing ways of seeing how data on proteins could become re-usable, could be reactivated as **collective property rather than the by-product of publication.**"

# The previous text can be easily transcribed to apply to software tool development

“This desire to reactivate software is widespread, and (*you*) are not alone in wanting to show that ‘far from being places where code goes to die’ (*your paper*), **such code collections can be re-used as valuable tools that would be difficult to obtain in any other way.**”

“In attempting to reactivate sedimented code in order to enable its re-use, their first step was ...”

“... they are experiments in sharing, in furnishing ways of seeing how code and tools could become re-usable, could be reactivated as **collective property rather than the by-product of publication.**”

# A field guide to open science for the newly initiated

Preprint

## NOT PEER-REVIEWED

"PeerJ Preprints" is a venue for early communication or feedback before peer review. Data may be preliminary. [Learn more about preprints](#) or [browse peer-reviewed articles instead](#).

## Do you speak open science? Resources and tips to learn the language

Science and Medical Education

Paola Masuzzo<sup>1,2</sup>, Lennart Martens<sup>1,2</sup>

January 3, 2017

> Author and article information

▾ Abstract

The internet era, large-scale computing and storage resources, mobile devices, social media, and their high uptake among different groups of people, have all deeply changed the way knowledge is created, communicated, and further deployed. These advances have enabled a radical transformation of the practice of science, which is now more open, more global and collaborative, and closer to society than ever. Open science has therefore become an increasingly important topic. Moreover, as open science is actively pursued by several high-profile funders and institutions, it has fast become a crucial matter to all researchers. However, because this widespread interest in open science has emerged relatively recently, its definition and implementation are constantly shifting and evolving, sometimes leaving researchers in doubt about how to adopt open science, and which are the best practices to follow.

Enter your institution

To find colleagues at PeerJ

Enter to search

Download preprint as...

Follow for updates

### Tools & info

Citations in Google Scholar

Add feedback 5

Ask questions

Add links

Visitors 3,228 [click for details](#)

Views 4,456

Downloads 510

### Outline

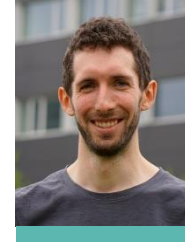
#### PeerJ Job Listings [beta]

List & find academic jobs on PeerJ for free.

[Learn more >](#)

Get PeerJ content alerts





Comp  
omics



[www.compomics.com](http://www.compomics.com)  
[compomics.github.io](https://compomics.github.io)