From the lab to Jupyter

A brief history of computational notebooks

Émilien Schultz (CREST - IP Paris)

2024-02-03

What I want to do

- A too short history of Jupyter's notebooks
- To plea for a better knowledge of scientific softwares' history.

Where are (our) stories of scientific software ?

I mean, outside the open research devroom...

Software is everywhere in research (OS and $\overline{OS}$):

- bespoken code
- small packages
- dedicated softwares
- international stars

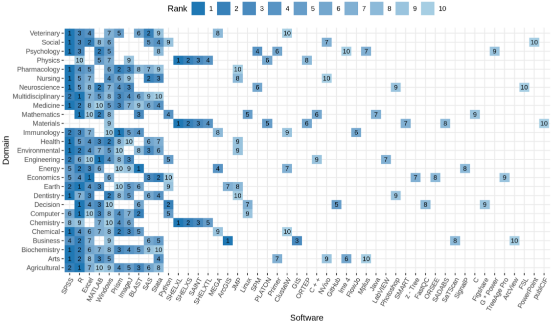


Figure 7 **Top 10 software per domain.** Higher rank within the domain is represented by darker color. The number on the tile gives the rank within the domain. Software with rank higher than 10, are excluded from the plot to improve readability. Software are ordered by rank over all domains left to right.
Full-size DOI: 10.7717/peerj-cs.835/fig-7

Schindler, David, et al. "The role of software in science: a knowledge graph-based analysis of software mentions in PubMed Central." PeerJ Computer Science 8 (2022): e835.

When social sciences look (rarely) at scientific software[1]

- Specific dynamics
  - open-ended & uncertain aims
  - researchers are (usually) non-specialist developers
  - effects of funding constraints
- Specific consequences
  - code brittleness
  - intertwined with scientific activity
  - specialization of some researchers
- Led to specific *software journeys*

[1] *Spencer, Matt. ”Brittleness and bureaucracy: software as a material for science.” Perspectives on Science 23.4 (2015): 466-484.*

Taking a step back : a lot of open questions

- How to tell the stories of scientific software ?
  - What are the different software journeys, especially in OS ?
  - What are the steps composing such journeys and condition of evolution ?
- How much are intertwined the histories of open source & academia ?
  - relation between open source & open science ?
  - relation between academics and software engineer ?[2]

ℹ Two Bits: The Cultural Significance of Free Software, Kelty, 2008, p.132

> *The fact that UNIX spread first to university computer-science departments, and not to businesses government, or nongovernmental organizations, meant that it also became part of the core pedagogical practices of a generation of programmers and computer scientists [...]*

[2] Woolston, Chris. "Why science needs more research software engineers." Nature (2022).

Put this idea to work with the case of Jupyter's notebooks

Innovation from research evolving to a worldwide infrastructure of data science

- Released in 2012, spreaded everywhere
    - ACM Award in 2017
- A perfect viewpoint[a] of :
    - emergence
    - incremental abstraction
    - diffusion within and outside

[a] Long version of this history in French

Caveat : not here to advocate about Jupyter notebooks

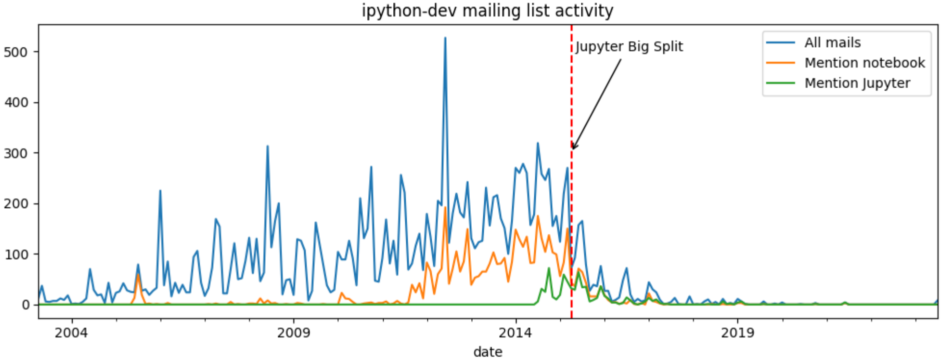
Postulate : you know what Jupyter's computationnal notebooks are, and you have your opinion about them[3]

I don't like notebooks.- Joel Grus (Allen Institute for Artificial Intelligence)

[3] I = ipython and Jupyter user

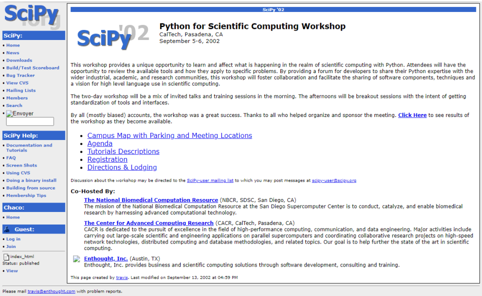
Oversight of the history

First a PhD student, then IPython, next notebooks, and finally Jupyter



Let's dive in. First, the context early 2000[4]

- 90' Free software achievements
- Litterate programming paradigm is around (Knuth, 1984)
- Specialized interactive scientific softwares (Maple, 1992, Mathematica, 1988)
- A nascent scientific Python community
    - First SciPy workshop in 2002



Python for Scientific Computing Workshop
CalTech, Pasadena, CA
September 5-6, 2002

This workshop provides a unique opportunity to learn and affect what is happening in the realm of scientific computing with Python. Attendees will have the opportunity to review the available tools and how they apply to specific problems. By providing a forum for developers to share their Python expertise with the wider industrial, academic, and research communities, this workshop will foster collaboration and facilitate the sharing of software components, techniques and a vision for high level language use in scientific computing.

The two-day workshop will be a mix of invited talks and training sessions in the morning. The afternoons will be breakout sessions with the intent of getting standardization of tools and interfaces.

By all (mostly biased) accounts, the workshop was a great success. Thanks to all who helped organize and sponsor the meeting. **Click Here** to see results of the workshop as they become available.

- Campus Map with Parking and Meeting Locations
- Agenda
- Tutorials Descriptions
- Registration
- Directions & Lodging

Discussion about the workshop may be directed to the SciPy-user mailing list to which you may post messages at scipy-user@scipy.org

Co-Hosted By:

The National Biomedical Computation Resource (NBCR, SDSC, San Diego, CA)
The mission of the National Biomedical Computation Resource at the San Diego Supercomputer Center is to conduct, catalyze, and enable biomedical research by harnessing advanced computational technology.

The Center for Advanced Computing Research (CACR, CalTech, Pasadena, CA)
CACR is dedicated to the pursuit of excellence in the field of high-performance computing, communication, and data engineering. Major activities include carrying out large-scale scientific and engineering applications on parallel supercomputers and coordinating collaborative research projects on high-speed network technologies, distributed computing and database methodologies, and related topics. Our goal is to help further the state of the art in scientific computing.

Enthought, Inc. (Austin, TX)
Enthought, Inc. provides business and scientific computing solutions through software development, consulting and training.

This page created by travis. Last modified on September 13, 2002 at 04:59 PM

Please mail travis@enthought.com with problem reports.

[4] The open source community keep track of some element of its history, i.e. William Horton's talk

Jupyter started as a small hack

- Fernando Pérez started IPython in 2001
  - during his PhD in particles physics
  - "*a simple personal fix for a problem in my own workflow*"
- Grounded in his researcher's common sense
  - A taste for interactivity in scientific programming

ℹ Fernando Pérez, 2012

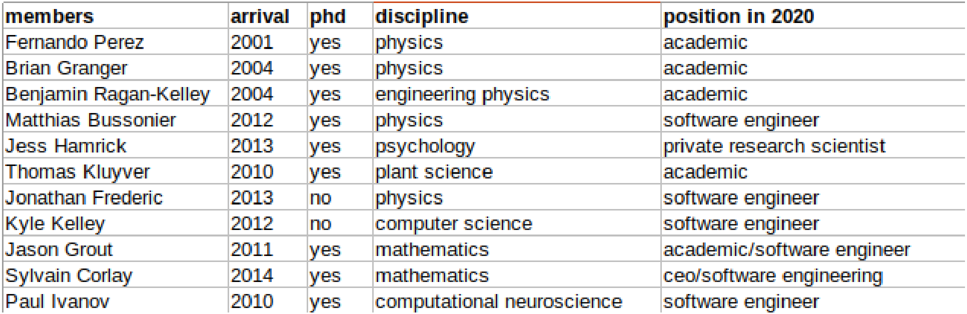
If you typed Python in the command line, you got a, an interactive shell, it was a very, very primitive and it didn't allow me to do the kinds of things that were very natural in interactive scientific workflows with tools like IDL or Mathematica that I used heavily or Matlab or Maple that other used which was simply to type a bit of code, see the results right there, open a plot, look at the files on, on the file system, et cetera.

The SciPy community as a amplifier

- Enthousiastic reception
  - Enthought hosted IPython on its webpage
- Increasing support
  - Feedback and contributors
  - Brian Granger, Fernando's friend, jumped in 2004
- Securing the financial possibility to carry on
  - Possibility of a postdoc
  - Support of a team leader in Berkeley (moved there in 2008)

> Fernando Pérez, 2021
>
> *Very importantly, though, there were people at UC Berkeley early on who supported me when I was still a postdoctoral scholar in Colorado doing more traditional applied mathematics research with Python tools. I wasn't invested enough in the purely applied mathematics community to make a career just out of that. People at UC Berkeley that I connected to because of the Python community offered me a team and, eventually, a job.*

Led IPython to be grounded in academia



members	arrival	phd	discipline	position in 2020
Fernando Perez	2001	yes	physics	academic
Brian Granger	2004	yes	physics	academic
Benjamin Ragan-Kelley	2004	yes	engineering physics	academic
Matthias Bussonier	2012	yes	physics	software engineer
Jess Hamrick	2013	yes	psychology	private research scientist
Thomas Kluyver	2010	yes	plant science	academic
Jonathan Frederic	2013	no	physics	software engineer
Kyle Kelley	2012	no	computer science	software engineer
Jason Grout	2011	yes	mathematics	academic/software engineer
Sylvain Corlay	2014	yes	mathematics	ceo/software engineering
Paul Ivanov	2010	yes	computational neuroscience	software engineer

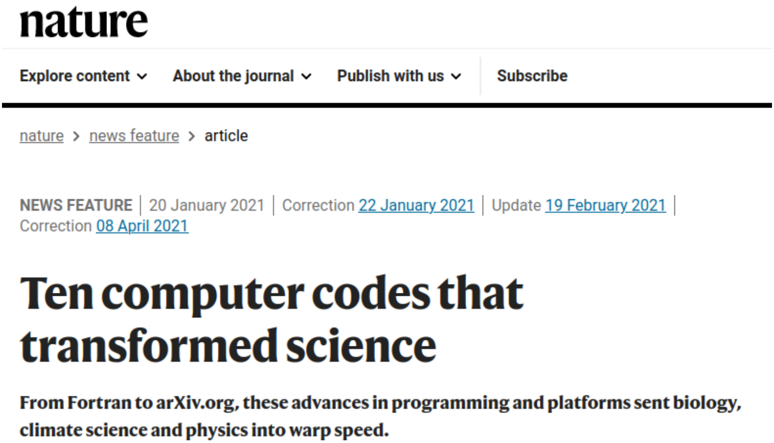
Notebooks were (just) a late feature of IPython (2012)

- Growth of the IPython project
  - Adding new features
  - New core contributors joining the project
  - Early private investment (Google, Boomberg, etc.)
- 5 failed attempts of notebooks between 2005 and 2011
  - Some tech not available (web sockets)

Fernando Pérez, 2012

On December 21 2011, we released IPython 0.12 after an intense 4 1/2 months of development. Along with a number of new features and bug fixes, the main highlight of this release is our new browser-based interactive notebook: an environment that retains all the features of the familiar console-based IPython but provides a cell-based execution workflow and can contain not only code but any element a modern browser can display [...] ***For the IPython project this was a major milestone, as we had wanted for years to have such a system, and it has generated a fair amount of interest online.***

Worked pretty well



NEWS FEATURE | 20 January 2021 | Correction 22 January 2021 | Update 19 February 2021 | Correction 08 April 2021

Ten computer codes that transformed science

From Fortran to arXiv.org, these advances in programming and platforms sent biology, climate science and physics into warp speed.

April 16, 2015 : Jupyter *Big Split*^{TM}

Evolution to a broader open source interactive scientific computing framework grounded on open science values

- The effect of two dynamics :
  - Abstracting from the Python community
    - “*Those langages are not ennemies, the ennemies are closed science*”
  - Strengthening software engineering good practices
- Backed with a 6 millions dollars grant
  - Helmsley Trust, Gordon and Betty Moore & Alfred P. Sloan Foundation

ℹ Fernando Pérez & Brian Granger, grant application, 2015

> ***we have quite ambitions plans for the future*** *[...] Project Jupyter’s mission is to create open source tools for interactive scientific computing and data science in research, education and industry, with an emphasis on usability, collaboration and reproducibility*

Jupyter's community overflowed academia

- Worldwide adoption
  - notebooks became standard of data science
  - Integration in a diversity of services (Colab...) and third-party (VS Code ...)
- But still a strong kernel of academic users and contributors

> ℹ Fernando Pérez, 2021
>
> *This community is not accidental: the core Jupyter team has invested significant effort into welcoming new contributors, helping users, planning and running community events (Jupyter CommunityWorkshops24, JupyterDays and JupyterCon25), and training and mentoring junior developers and designers.*

Let's stop here in this *work in progress*

A lot of open questions :

- Who are the current users of computationnal notebooks ?
- How are researchers using them ?
- How did the Jupyter ecosystem expanded ?
- ...

And, at this point, **does Jupyter still is a scientific software ?**

Wrapping up : (historical) documentation is important

Scientific softwares are crucial research instruments.

- Need to take their dynamics seriously
- Of course, computationnal notebooks
  - indicative of scientific programming evolutions
  - ongoing project NOOS[5]
  - a Github repo for collective archive
- But also all the others
  - Rstudio
  - Gephi
  - ...

[5] Célya Gruson-Daniel, Mariannig Le Béchec, Clémence Lascombes