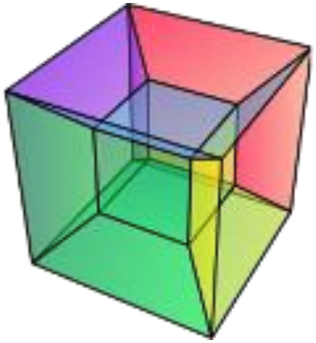
HyperSpy Community & Github

Eric Prestat

University of Manchester and SuperSTEM Laboratory



- HyperSpy on the web
- What is Github? Why is it so popular?
- What makes the HyperSpy community?

HyperSpy documentation/community support

- Main website:
  - http://hyperspy.org/
- Documentation:
  - http://hyperspy.readthedocs.io/en/stable/
- Users list:
  - https://groups.google.com/forum/#!forum/hyperspy-users
  - Discuss/ask use case of HyperSpy
- Gitter: online chat room
  - https://gitter.im/hyperspy/hyperspy

HyperSpy demos

- Source code of the demo on github
  - https://github.com/hyperspy/hyperspy-demos
- Non-interactive version (using nbviewer)
  - http://nbviewer.jupyter.org/github/hyperspy/hyperspy-demos/tree/master/
  - Github now displays nicely the notebook, nbviewer is not really necessary anymore
- Interactive version (using mybinder)
  - https://mybinder.org/v2/gh/hyperspy/hyperspy-demos/master
  - Run demos online (on a remote server) without any installation
  - May be a bit slow…

HyperSpy development

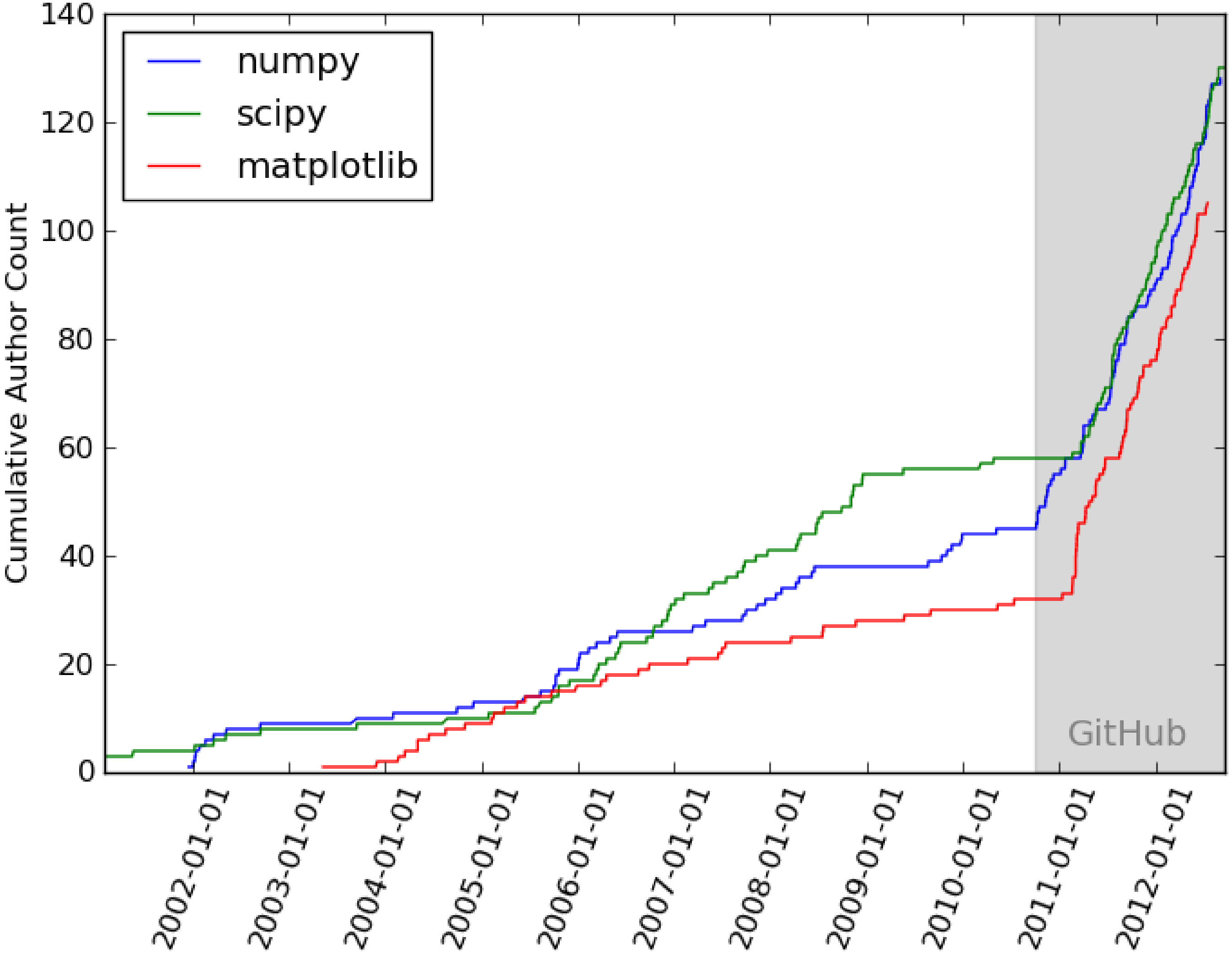
- Development site on Github
  - https://github.com/hyperspy/hyperspy
  - Issues tracker: report bug, propose new features and other business
  - Pull requests: discuss the merge of any changes with the upstream branches
- Developer guide
  - http://hyperspy.readthedocs.io/en/stable/dev_guide.html

HyperSpy is built on many python libraries

- Strong scientist user base
- The python scientific stack is huge!
  - Generic array computation: Numpy, scipy
  - More specialise libraries: scikit-image, scikit-learn
  - Visualisation: matplotlib & others
  - High-performance computing: Cython, numba, dask
  - Interactivity/user interface: jupyter, ipython, spyder
- One big strength of python is his large and dynamic community
  - Progress is fast, which is great!
  - Sometimes, it breaks because it move quickly…

https://www.scipy.org/about.html

The github effect



From Pythonic perambulations: Why Python is the last language you'll have to learn: https://jakevdp.github.io/blog/2012/09/20/why-python-is-the-last/

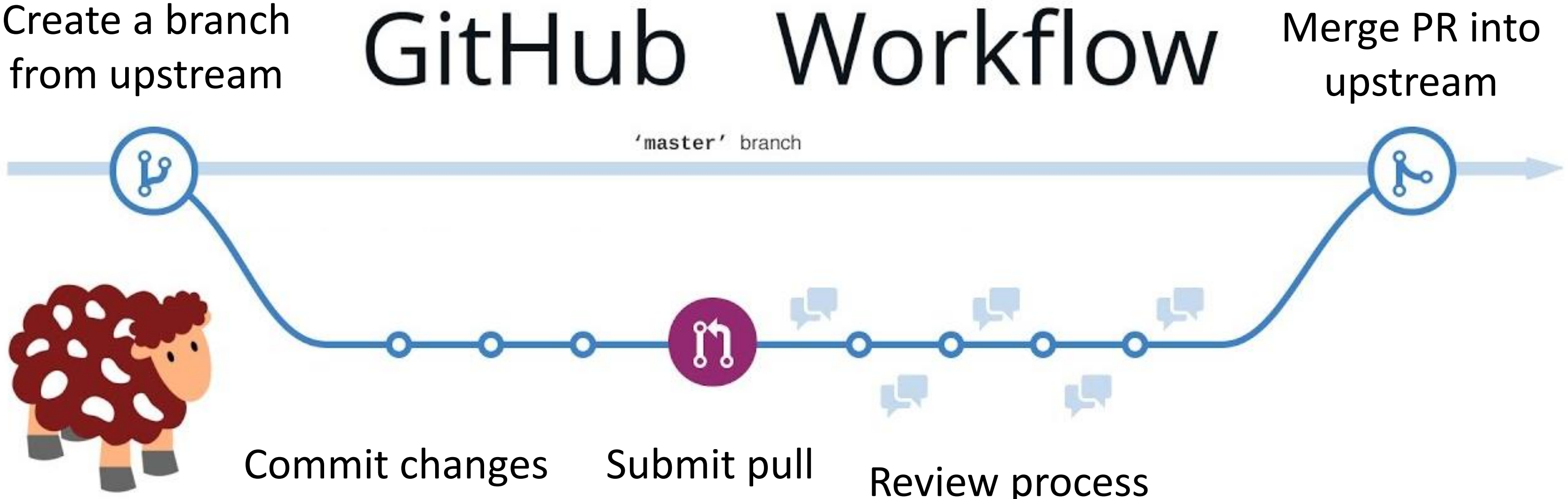
What is github?

- At the heart of Github is Git
  - A version control systems: manages and stores revisions of projects
  - Git is a distributed version control systems
    - Each contributor has is own *remote* (online) repository
    - Code is merged in the *upstream* repository

- GitHub is a Git repository hosting service with its own features
  - Web-based graphical interface
  - provides a centralised place where people discuss the changes
  - “Social coding”
    - Open discussion in the issue tracker or PR submission, wiki, etc

Short video explaining github: https://www.youtube.com/watch?v=w3jLJU7DT5E

How to use github?



Create a branch from upstream

Commit changes to one of your branch

Submit pull request (PR)

Review process

Merge PR into upstream

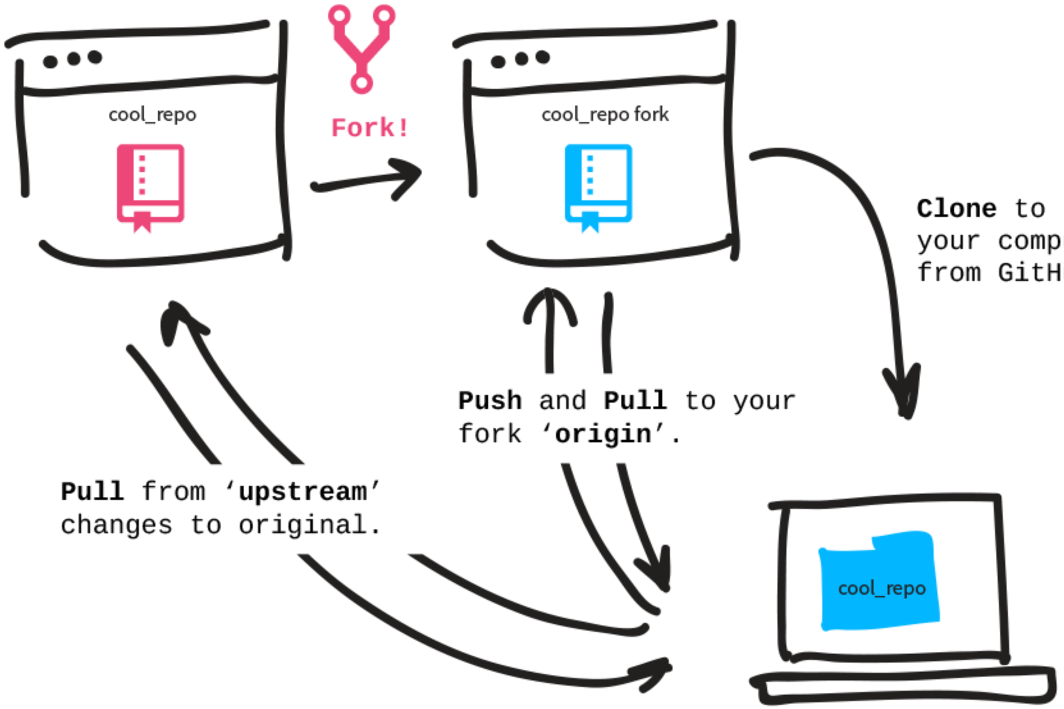
How to use github?

REMOTE

Someone else's repository.

REMOTE

Your fork of the repository.



Clone to your computer from GitHub.

Push and **Pull** to your fork '**origin**'.

Pull from '**upstream**' changes to original.

LOCAL

Use your computer's **terminal** to talk to two repositories via **two remotes** to the GitHub servers.

Contribute to an open source project before git/github

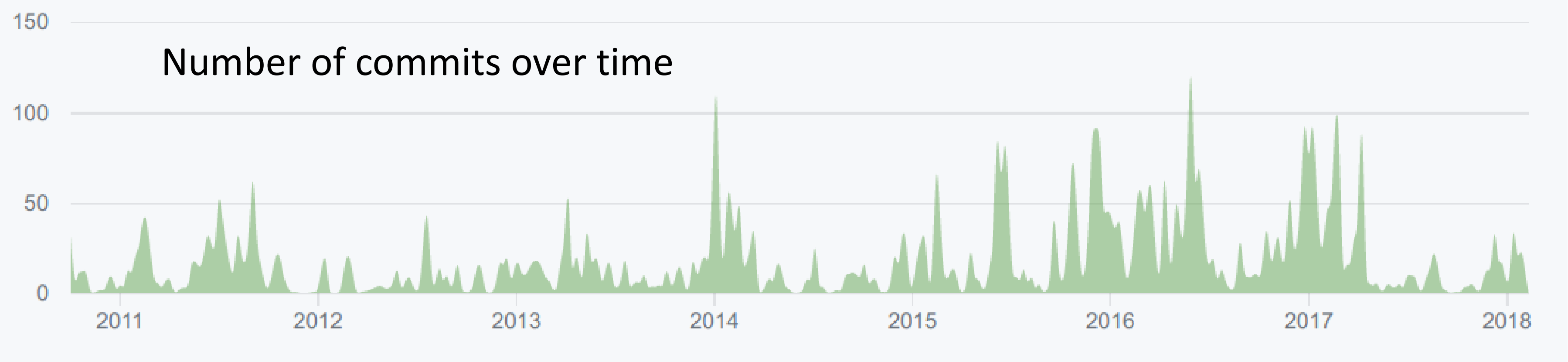
1. manually download the project’s source code
2. make your changes locally
3. create a list of changes called a ‘patch’ and then e-mail the patch to the project’s maintainer
4. The maintainer would then have to evaluate this patch, possibly sent by a total stranger and decide whether to merge the changes.

Why is github so useful/popular?

- github web interface makes things easier:
  - Share code with other developers and the public
  - Manage issues
  - Manage pull request (code comparison and review)
  - Maintain code
- *Continuous integration*
  - For each PR, the code is tested automatically against a suite of tests using external services (travis, appveyor, etc.)
- A *distributed community* of contributors

HyperSpy community

- Code contributors
  - 33 contributors in total from many different labs
  - A few contributors change jobs
    - Their github profile may have been useful for their successful application
  - ~ 10 of them is a one off contribution
  - But most importantly, we have regular contributors



- Many more people following the gitter chat or the google user list contribute to the HyperSpy community
  - Bug report, feedback, complain, etc.

Common issues with research software development

- Many publications present interesting data analysis workflow or methodological development
- These are generally not easily accessible
  - proprietary licence
  - not easily to install or to run (platform dependent, etc.)
  - not supported/up to date anymore
- May be difficult to maintain for the developer
- Implementation need to be generic and robust
- Rely on a specific research group (most of the time by one or a few PhD student or a post-doc..)
  - Not sustainable

What makes HyperSpy today?

- Use tools and development practises which have proven to be successful for open source project
- HyperSpy is not lead by a research group/institution
- HyperSpy is supported by its own community
- Peer-reviewed and open-source development
- HyperSpy is becoming a mature library
  - API fairly stable
  - doesn't break as much as before
- HyperSpy can be integrated easily in other software
  - offer a powerful platform for the development of other libraries
  - HyperSpy will be split in the near future
  - Easier, faster implementation of new features