Open source to change the culture of science

Matthew Brett

Grant team

Matthew Brett (Nibabel developer)

Chris Markiewicz (Nibabel lead maintainer)

Oscar Esteban (developer of Nitransforms tool)

Democratizing brain imaging through code

We want to build a culture in which it is ordinary for scientists to contribute to their tools.

If we succeed, we will go a long way to solving the chronic and severe problems of monoculture in open source.

To do this, we need to build use and contribution down into the deepest possible foundation in education.

The user, the developer, the scientist

- Old model: developers develop, scientists use.
- The new model: the developer is the scientist.
- The *practitioner build*.
- The *user-contributor*.

https://asterisk.dynevor.org/how-do-the-foundations-get-built.html

Code as foundation

> Data science is an approach to data analysis with a foundation in code and algorithms.

https://matthew-brett.github.io/cfd2020/intro/what-is-data-science.html

Code to learn

> What I cannot create, I do not understand

Richard Feynman - found on his blackboard after his death

Code as explanation

https://nipy.org/nibabel/image_orientation.html

Software for understanding

> The purpose of NIPY is to make it easier to do better brain imaging research. We believe that neuroscience ideas and analysis ideas develop together. Good ideas come from understanding; understanding comes from clarity, and clarity must come from well-designed teaching materials and well-designed software. The software must be designed as a natural extension of the underlying ideas.

NIPY mission statement

The brain imaging landscape

Practitioner-build, open-source, but little user-contribution.

- AFNI - C/C++
- FSL - C/C++/some Python/Tcl/tk
- SPM - MATLAB

Nibabel

- A base library for reading and manipulating brain images.
- Dependency of all popular Python brain imaging libraries.
- https://github.com/nipy/nibabel

The way of code

```python
import nibabel as nib

# Create brain image object
img = nib.load('my_image.nii.gz')

# Read data from disk as an array
data = img.get_fdata()
```

If time, a notebook.

Making code ordinary in teaching

- As example of reproducible science - Millman et al. (2018) - Frontiers article.
- Full imaging course, combined with reproducible science: https://bic-berkeley.github.io/psych-214-fall-2016/

Changing the culture

- Fundamental courses for which code is ordinary.
- Make sure that the library scales with the task.
- Build in the mechanisms for contribution (they are also the mechanisms for collaboration).
- Find and mentor the user-contributors and practitioner builders of the next generation.

Our grant

- Build up the library so it can scale.
- Build and improve and distribute courses that integrate code and collaboration.

The end

Materials at https://github.com/matthew-brett/czi-nibabel, in the `talk` directory.

References

Millman, K. Jarrod, Matthew Brett, Ross Barnowski, and Jean-Baptiste Poline. 2018. "Teaching Computational Reproducibility for Neuroimaging." *Frontiers in Neuroscience*.