

Reproducible software vs. reproducible research

Fernando Pérez

<http://fperez.org>

Fernando.Perez@berkeley.edu

Helen Wills Neuroscience Institute, UC Berkeley

AAAS Annual Meeting,

The Digitization of Science: Reproducibility and Interdisciplinary Knowledge Transfer

Washington, DC

Feb 19, 2011

Outline

- 1 A contrast of cultures
- 2 Technical ideas: tools matter
- 3 Incentives and rewards: changing our practices

Outline

- 1 A contrast of cultures
- 2 Technical ideas: tools matter
- 3 Incentives and rewards: changing our practices

Who am I?

Background

- Particle physics (theory/computation): numerical QCD
- Applied mathematics: algorithm development for PDEs
- Neuroscience: algorithms and tools for brain imaging

A common thread: computational tools

- IPython: interactive Python
- Matplotlib: visualization
- Numpy: numerics
- Scipy: scientific algorithms
- Nipy: neuroimaging tools

What does it take to get reproducible research *results*?

Reproducible research practices!

Reproducibility at publication time?

It's already too late.

Learn from a community (open source) where
reproducibility is an everyday practice
(by necessity)

What does it take to get reproducible research *results*?

Reproducible research practices!

Reproducibility at publication time?

It's already too late.

Learn from a community (open source) where
reproducibility is an everyday practice
(by necessity)

What does it take to get reproducible research *results*?

Reproducible research practices!

Reproducibility at publication time?

It's already too late.

Learn from a community (open source) where
reproducibility is an everyday practice
(by necessity)



ELSEVIER

Available online at www.sciencedirect.com



ScienceDirect

Appl. Comput. Harmon. Anal. 24 (2008) 354–377

**Applied and
Computational
Harmonic Analysis**

www.elsevier.com/locate/acha

Fast adaptive algorithms in the non-standard form for multidimensional problems [☆]

Gregory Beylkin *, Vani Cheruvu, Fernando Pérez

Department of Applied Mathematics, University of Colorado, Boulder, CO 80309-0526, USA

Received 6 June 2007; accepted 2 August 2007

Available online 14 August 2007

Communicated by Vladimir Rokhlin

Abstract

We present a fast, adaptive multiresolution algorithm for applying integral operators with a wide class of radially symmetric kernels in dimensions one, two and three. This algorithm is made efficient by the use of separated representations of the kernel. We discuss operators of the class $(-\Delta + \mu^2 I)^{-\alpha}$, where $\mu \geq 0$ and $0 < \alpha < 3/2$, and illustrate the algorithm for the Poisson and Schrödinger equations in dimension three. The same algorithm may be used for all operators with radially symmetric kernels

Pages of algorithmia as equations or vague methods descriptions

a periodic analogue of the Hilbert transform. In order to find its representation in multiwavelet bases, we compute

$$r_{ii'}^{j;l} = 2^{-j} \int_{-1}^1 K(2^{-j}(x+l)) \Phi_{ii'}(x) dx = 2^{-j} \int_{-1}^1 \cot(\pi 2^{-j}(x+l)) \Phi_{ii'}(x) dx, \quad (12)$$

where $\Phi_{ii'}(x)$, $i, i' = 0, \dots, k-1$ are cross-correlation functions described in Appendix A.4 and $l = 0, \pm 1, \pm 2, \dots, 2^j - 1$. We compute $r_{ii'}^{j;l}$ using the convergent integrals

$$r_{ii'}^{j;l} = 2^{-j} \sum_{k=i'-i}^{i'+i} c_{ii'}^k \int_0^1 \Phi_{k,0}^+(x) (\cot(\pi 2^{-j}(x+l)) + (-1)^{i+i'} \cot(\pi 2^{-j}(-x+l))) dx,$$

where $\Phi_{k,0}^+$ is a polynomial described in Appendix A.4. In our numerical experiment, we apply (11) to the periodic function on $[0, 1]$,

$$f(x) = \sum_{k \in \mathbb{Z}} e^{-a(x+k-1/2)^2},$$

Tables and pretty figures. Now in color!

Table 1
Results from evaluating (13) with our algorithm

p	Scales	N_{blocks}	ϵ	E_2
5	[2,3,4]	8	10^{-3}	1.5×10^{-4}
8	[2,4,5]	12	10^{-6}	1.3×10^{-7}
11	[2,4,5]	14	10^{-9}	1.1×10^{-10}
14	[3,4,5]	16	10^{-12}	4.4×10^{-13}

Notes. The order of the basis p is adjusted as a function of the requested precision ϵ . The second column indicates scales present in the adaptive tree for the input. The third column shows the total number of blocks of coefficients in this tree. The last column (E_2) shows the actual error of the computed solution in the ℓ^2 norm.

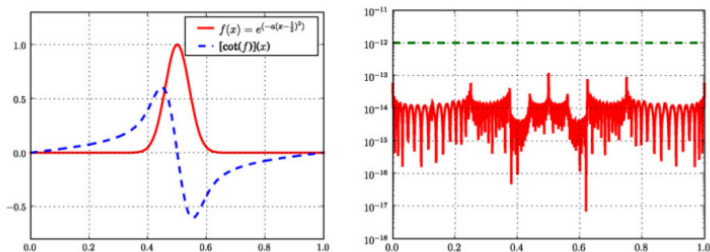


Fig. 3. Results of applying the cotangent kernel to a periodized Gaussian using basis of order $p = 14$ (the last row in Table 1). The pointwise error is shown on the right for a requested accuracy of $\epsilon = 10^{-12}$.

Chance of reproducing results for third parties?

$\mathcal{O}(10^{-\text{something very big}})$

[STRFPak](#)[Challenge](#)

STRFlab

Spatio-temporal receptive field lab

Download STRFlab

Please register to download STRFlab:

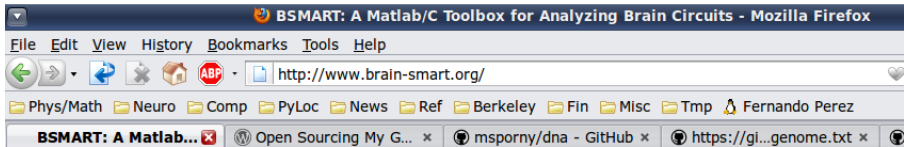
Comments

Sign up for mailing list:

☐

Download

Zip file/tarball dumps



BSMART: A Matlab/C Toolbox for Analyzing Brain Circuits

BSMART, an acronym of Brain-System for Multivariate AutoRegressive Timeseries, is an open-source software package for analyzing that was born out of a collaborative research effort between Dr. Hualou Liang at Drexel University, Dr. Steven Bressler at Florida Atlantic University of Florida. BSMART can be applied to a wide variety of neuroelectromagnetic phenomena, including EEG. A unique feature of the BSMART package is Granger causality that can be used to assess causal influences and directions of flow.

The backbone of the BSMART project is Multivariate AutoRegressive (MAR) analysis that has been long developed for statistical analysis at different time scales. Based upon a MAR model, a plethora of spectral quantities such as auto power, partial power, coherence and Granger causality can be immediately derived. The approach has been fruitfully used to characterize, with high resolution, functional relations within large scale brain networks.

The BSMART is currently undergoing beta test, freely available under the GNU public license ([download BSMART](#)). It is supported by the National Institute of Neurological Disorders and Stroke (NINDS) through the NIH Neuroinformatics / Human Brain Projects.

The BSMART is described in:

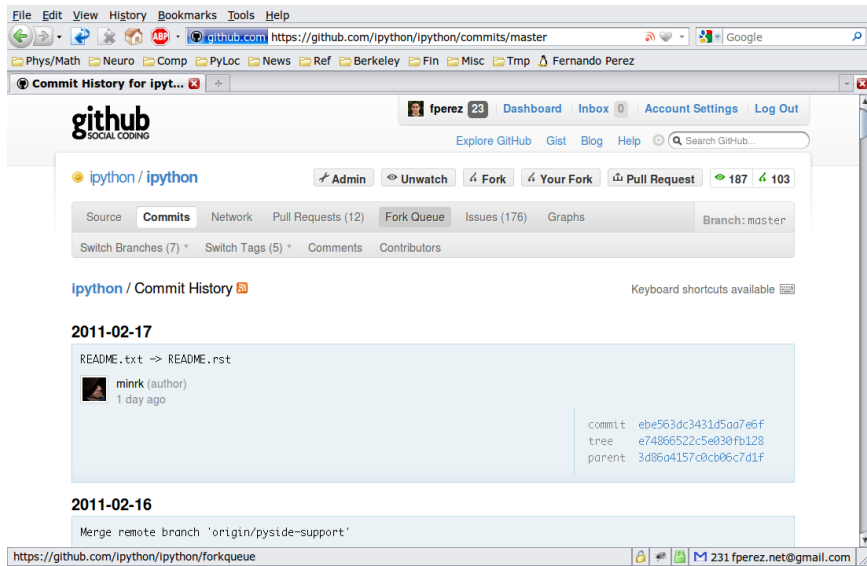
Jie Cui, Lei Xu, Steven L. Bressler, Mingzhou Ding, Hualou Liang, BSMART: a Matlab/C toolbox for analyzing neural time series, *Neural Networks, Special Issue on Neuroinformatics*, 21:1094 - 1104, 2008. ([doi](#))

Please refer to this article when publishing results obtained from the BSMART toolbox. For requesting comments, please contact Hualou Liang at <http://www.brain-smart.org/download/bsmart0p5b105.zip>

Contrast: FOSS better than scientific research?

FOSS: Free and Open Source Software

Public distributed version control: provenance tracking



The screenshot shows a web browser window displaying the GitHub repository for `ipython/ipython`. The browser's address bar shows the URL `https://github.com/ipython/ipython/commits/master`. The repository page includes navigation links like `Dashboard`, `Inbox`, `Account Settings`, and `Log Out`. The commit history section shows a commit by `minrk` on 2011-02-17, with a message `README.txt -> README.rst`. Below this, a commit from 2011-02-16 is partially visible with the message `Merge remote branch 'origin/pside-support'`. The bottom of the browser window shows the address bar with `https://github.com/ipython/ipython/forkqueue` and a taskbar with an email icon and the address `231 fperez.net@gmail.com`.

File Edit View History Bookmarks Tools Help

github.com https://github.com/ipython/ipython/commits/master

Phys/Math Neuro Comp PyLoc News Ref Berkeley Fin Misc Tmp Fernando Perez

Commit History for ipyt...

github SOCIAL CODING

fperez 23 Dashboard Inbox 0 Account Settings Log Out

Explore GitHub Gist Blog Help Search GitHub...

ipython / ipython Admin Unwatch Fork Your Fork Pull Request 187 103

Source Commits Network Pull Requests (12) Fork Queue Issues (176) Graphs Branch: master

Switch Branches (7) Switch Tags (5) Comments Contributors

ipython / Commit History

Keyboard shortcuts available

2011-02-17

README.txt -> README.rst

minrk (author)
1 day ago

commit ebe563dc3431d5aa7e6f
tree e74866522c5e030fb128
parent 3d86a4157c0cb06c7d1f


2011-02-16

Merge remote branch 'origin/pside-support'

https://github.com/ipython/ipython/forkqueue

231 fperez.net@gmail.com

Pull requests: ongoing peer review



fperez 23 | [Dashboard](#) | [Inbox 0](#) | [Account Settings](#) | [Log Out](#)

[Explore GitHub](#) | [Gist](#) | [Blog](#) | [Help](#) |


ipython / ipython

[Admin](#) | [Unwatch](#) | [Fork](#) | [Your Fork](#) | [Pull Request](#) | [187](#) | [103](#)

[Source](#) | [Commits](#) | [Network](#) | **[Pull Requests \(12\)](#)** | [Fork Queue](#) | [Issues \(176\)](#) | [Graphs](#) | [Branch: master](#)

Open takluyver wants someone to merge 9 commits into `ipython:master` from `takluyver:issue-245` #261

[Discussion](#) | [Commits <> 9](#) | [Diff >= 11](#)



takluyver opened this pull request February 05, 2011


Adapt magic commands to new history system.

This grew from issue [ipython/ipython#245](#). Various magic commands weren't working properly with the new history system: %edit, %macro, and %hist.

Among various minor troubles, selecting a range of lines (%macro test 2-5) numbered from the beginning of the history, so didn't match up with the current line numbers. I've approached this by adding a session_offset attribute to the history manager. This has the added benefit that we no longer need to store a blank history entry so we can count lines from 1.

Along the way, I simplified and modernised parts of the code, including using `basestring` over `StringTypes` and `.isdigit()` over an equivalent regex.

Open
+ 151 additions
- 127 deletions
[All Pull Requests](#)



Pull requests: back and forth discussion



fperez started a discussion in the diff February 08, 2011

IPython/core/history.py

```
... @@ -77,6 +80,9 @@ class HistoryManager(object):  
77 80     # pre-processing.  This will allow users to retrieve the input just as  
78 81     # it was exactly typed in by the user, with %hist -r.  
79 82     self.input_hist_raw = []  
83 +  
84 +     # Offset so the first line of the current session is #1  
85 +     self.session_offset = -1
```

1



fperez repo collab

February 08, 2011

Since this is a new attribute, it should be listed at the class level, for details see: http://ipython.scipy.org/doc/nightly/html/development/coding_guide.html#attribute-declarations-for-objects

Add a line note

Branches: exploratory work with control

The screenshot shows the 'gitg - ipython (master)' application window. The top menu bar includes 'File', 'Edit', 'View', 'Repository', and 'Help'. Below the menu is a toolbar with 'History' and 'Commit' buttons. The 'Branch:' dropdown is set to 'master'. The main area displays a commit history table with columns for 'Subject', 'Author', and 'Date'. A vertical timeline on the left shows the sequence of commits and branches. The 'Details' tab is selected, showing information for the selected commit: SHA: 3d86a4157c0cb06c7d1fa12ee97bdd5c83f6e712, Author: epatters, Date: Wed 16 Feb 2011 11:22:17 AM EST, and Subject: Merge remote branch 'origin/pyside-support'. The parent commit is 604c8e1ef5e5bf07cf48a82eb2acc07b10b9a42b (Merge branch 'ready_cProfile' of https://github.com/tomspur/ipython) and the commit message is 12ee02612522e257a80fc803c082ce394030e853 (PySide fix: PySide's QByteArray constructor does not overload for unicode.).

Subject	Author	Date
master origin/master README.txt -> README.rst	MinRK	Thu 17 Feb 2011 12:03:09 PM
Merge remote branch 'origin/pyside-support'	epatters	Wed 16 Feb 2011 11:22:17 AM
Merge branch 'ready_cProfile' of https://github.com/tomspur/ipython	Thomas Spura	Wed 16 Feb 2011 12:21:07 AM
Merge branch 'maglc-examples'	Thomas Kluyver	Tue 15 Feb 2011 03:46:10 PM
Skip doctests where necessary.	Thomas Kluyver	Tue 15 Feb 2011 02:34:40 PM
Add example for %cpaste	Sathesh Chandra	Tue 15 Feb 2011 02:27:38 PM
Wrote example for 'colors' command	vankayala sowjanya	Tue 15 Feb 2011 02:12:44 PM
Wrote an example for 'pdef'	vankayala sowjanya	Tue 15 Feb 2011 02:07:59 PM
Tweaks to RST formatting.	Thomas Kluyver	Tue 15 Feb 2011 01:59:38 PM
BUG: Do not store class-specific state on TraitTypes since they may be shared through subcl.	Robert Kern	Tue 15 Feb 2011 12:55:37 PM
epatters-pyside-support PySide fix: PySide's QByteArray constructor does not overload for unicode.	epatters	Tue 15 Feb 2011 11:37:47 AM
Improved error message for Qt API switcher.	epatters	Tue 15 Feb 2011 11:21:35 AM
kernel sends reply on the right side of std<x>.flush	MinRK	Tue 15 Feb 2011 01:17:20 AM
fix+test %who_is type checking, skip %who doctests	MinRK	Mon 14 Feb 2011 04:36:30 PM

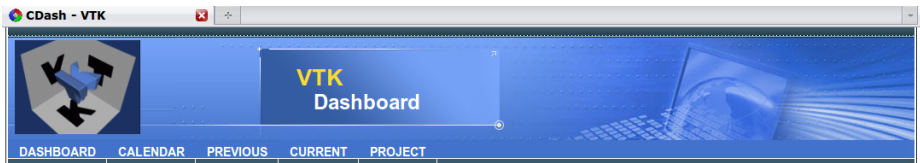
Details Tree

SHA: 3d86a4157c0cb06c7d1fa12ee97bdd5c83f6e712
Author: epatters
Date: Wed 16 Feb 2011 11:22:17 AM EST
Subject: **Merge remote branch 'origin/pyside-support'**
Parent: 604c8e1ef5e5bf07cf48a82eb2acc07b10b9a42b (Merge branch 'ready_cProfile' of https://github.com/tomspur/ipython)
12ee02612522e257a80fc803c082ce394030e853 (PySide fix: PySide's QByteArray constructor does not overload for unicode.)
Merge remote branch 'origin/pyside-support'

Loaded 3263 revisions in 0.45s

Automated tests: validation

The VTK Build Dashboard: immediate feedback



Nightly Expected

Site	Build Name	Update		Configure			Build			Test				Build Time
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min	
p90n03.pbm.ihost.com	AIX00F614-xLC	8	0.1	0	0	4.4	0	0	167.7	0	5	835	34.2	2011-02-19T04:01:58 EST
londonium.kitware	Arch-GCC-4.5-x86_64-debug	8	0.2	0	0	1	0	4	11.4	0	8	1356	45.4	2011-02-18T21:53:58 EST
londonium.kitware	Arch-GCC-4.5-x86_64-release	0	0.1	0	0	0.9	0	8	18.1	0	6	1357	27.9	2011-02-18T23:17:26 EST
amber1.kitware	Debian4-x64-gcc	8	0.4	0	0	1.2	0	0	51.4	0	1	518	10.2	2011-02-18T21:06:41 EST

Public bug trackers



sympy

Python library for symbolic mathematics

[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) [Source](#)

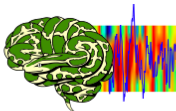
New issue [Advanced search](#) [Search tips](#)

							1 - 100 of 891 Next >	List Grid
ID ▼	Type ▼	Status ▼	Priority ▼	Milestone ▼	Owner ▼	Summary + Labels ▼		
2018	Defect	Accepted	Critical	Release0.7.0	smi...@gmail.com	terms vs factors NeedsBetterPatch smichr		
2126	Defect	Accepted	Critical	Release0.7.0	smi...@gmail.com	fix doctest or quality testing to recognize doctests NeedsBetterPatch smichr Testing		
2133	Defect	Started	Critical	Release0.7.0	matt...@gmail.com	Merge new polynomials manipulation module Polynomial NeedsBetterPatch mattpap NeedsReview		
2151	Defect	Accepted	Critical	Release0.7.0	Ronan.L...@gmail.com	BasicMeta.keep_sign Series		
1276	Defect	Started	High	Release0.7.0	----	solve(-1 + x**2 + 0.1111111111111111*(1.000000000000000 + 2.000000000000000*x)**2,x) fails EasyToFix Polynomial Solvers		
1721	Defect	Accepted	High	Release0.7.0	Ronan.L...@gmail.com	Rename class 'Real'		
1735	Defect	New	High	Release0.7.0	Ronan.L...@gmail.com	Rename .func attribute (.args too?)		
1919	Enhancement	Accepted	High	Release0.7.0	Vinzent.Steinberg	unify behavior of var() and symbols() NeedsReview smichr mattpap		
51	Enhancement	Started	Medium	Release0.7.0	matt...@gmail.com	RootOf for polynomial equations Polynomial NeedsReview mattpap		
326	Defect	Started	Medium	Release0.7.0	matt...@gmail.com	sympy.roots._sturm(...) hangs Polynomial EasyToFix NeedsReview mattpap		
527	Enhancement	Started	Medium	Release0.7.0	----	guessing what functions, like integrate, roots, factor, apart (and many more), should do with the given expression NeedsReview mattpap		

Documentation: Sphinx

Math, code and validated examples: literate programming

Nitime: time-series analysis for neuroscience



[Nitime Home](#) | [Nitime Documentation](#) » [Examples](#) »

[previous](#) | [next](#) | [modules](#) | [index](#)

Auditory processing in grasshoppers ¶

Extracting the average time-series from one signal, time-locked to the occurrence of some type of event in another signal is a very typical operation in the analysis of time-series from neuroscience experiments. Therefore, we have an additional example of this kind of analysis in [Event-related fMRI](#)

In the following code-snippet, we demonstrate the calculation of the spike-triggered average (STA). This is the average of the stimulus wave-form preceding the emission of a spike in the neuron and can be thought of as the stimulus 'preferred' by this neuron.

We start by importing the required modules:

```
import numpy as np

import nitime.timeseries as ts
import nitime.analysis as tsa
import nitime.viz as viz
```

Two data files are used in this example. The first contains the times of action potentials

Site Navigation

[Documentation](#)
[Development](#)
[News](#)

NIPY Community

[Community Home](#)
[NIPY Projects](#)
[Mailing List](#)
[License](#)

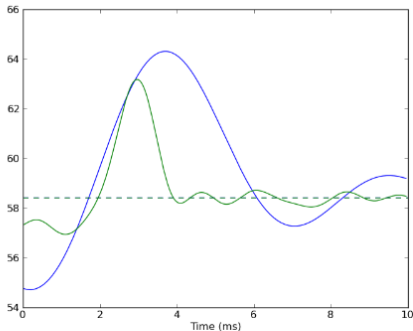
Previous topic

[Event-related fMRI](#)

Next topic

[Multi-taper coherence
estimation](#)

Docs with data, full code and references



`plt.show()` is called in order to display the figures

```
plt.show()
```

The data used in this example is also available on the [CRCNS data sharing web-site](#).

[Rokem2006] Ariel Rokem, Sebastian Watzl, Tim Gollisch, Martin Stemmler, Andreas V M Herz and Ines Samengo (2006). Spike-timing precision underlies the coding efficiency of auditory receptor neurons. *J Neurophysiol*, 95:2541-52

Example source code

You can download [the full source code of this example](#). This same script is also included in the Nitime source distribution under the `doc/examples/` directory.

Outline

- 1 A contrast of cultures
- 2 Technical ideas: tools matter
- 3 Incentives and rewards: changing our practices

Version control everywhere

Git: the tool you didn't know you needed

Reproducibility?

- Tracking and recreating every step of your work
- In the software world: it's called Version Control!

Git: an enabling technology. Use version control for everything

- Paper writing (**never** get `paper_v5_john.tex` by email again!)
- Grant writing
- Everyday research
- Teaching (**never** accept an emailed homework assignment again!)

Advantages of pervasive distributed version control

- Tracking of everyday results: a “time machine” view.
- Distributed backup.
- Explore lines of research/writing.
- Collaborate with colleagues.

Version control everywhere

Git: the tool you didn't know you needed

Reproducibility?

- Tracking and recreating every step of your work
- In the software world: it's called Version Control!

Git: an enabling technology. Use version control for everything

- Paper writing (**never** get `paper_v5_john.tex` by email again!)
- Grant writing
- Everyday research
- Teaching (**never** accept an emailed homework assignment again!)

Advantages of pervasive distributed version control

- Tracking of everyday results: a “time machine” view.
- Distributed backup.
- Explore lines of research/writing.
- Collaborate with colleagues.

Version control everywhere

Git: the tool you didn't know you needed

Reproducibility?

- Tracking and recreating every step of your work
- In the software world: it's called Version Control!

Git: an enabling technology. Use version control for everything

- Paper writing (**never** get `paper_v5_john.tex` by email again!)
- Grant writing
- Everyday research
- Teaching (**never** accept an emailed homework assignment again!)

Advantages of pervasive distributed version control

- Tracking of everyday results: a “time machine” view.
- Distributed backup.
- Explore lines of research/writing.
- Collaborate with colleagues.

Git: publish your genome!

<http://manu.sporny.org/2011/public-domain-genome>



fperez 23

[Dashboard](#)

[Inbox](#) 0

[Account Settings](#)

[Log Out](#)

[Explore GitHub](#)

[Gist](#)

[Blog](#)

[Help](#)

msporny / dna

Watch

Fork

229

26

[Source](#)

[Commits](#)

[Network](#)

[Pull Requests \(1\)](#)

[Issues \(8\)](#)

[Graphs](#)

Branch: master

Graph

[Members](#)

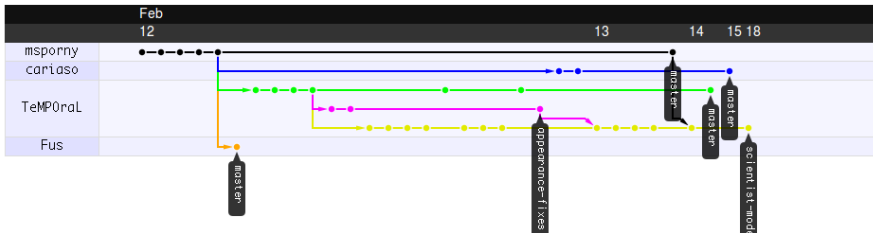
The dna network graph

Keyboard shortcuts available

All branches in the network using msporny/dna as the reference point. [Read our blog post about how it works.](#)

[Show Help](#)

Last updated: 1 day ago



Outline

- 1 A contrast of cultures
- 2 Technical ideas: tools matter
- 3 Incentives and rewards: changing our practices

Incentives and rewards

In Open Source

- Individual attribution in commit logs.
- Volunteers find reward in community.
- For some, it's part of their job.
- No hidden work before “publication”: the **process** is open.

Academia: a naïve transplant won't work

- Success \iff individual authorship.
- Fears of scooping from open development.
- Low/no requirements from journals
 - But changing! E.g. *Biostatistics*, *Open Research Computation*.
- Similarly for funding agencies.
 - Also changing: new NSF data management requirements.

Adopt a *habit* of reproducibility

Make version control as routine as email

- Git for your next [grant](#)
- Git in your next in-house [research project](#).
- Disk is cheap! Separate repositories for:
 - [Libraries](#): automated tests and docs *during the development process*.
 - [In-house tools](#) shared across project but of less generic use.
 - [Project/dataset](#) specific repositories.
- **Write** your next [paper](#) with a repository that can produce all results/figures.
- **Publish** your next [paper](#) with the code/data repository for it
 - Properly licensed, see V. Stodden's standard.

Use your influence to improve the situation

FINAL NIH STATEMENT ON SHARING RESEARCH DATA

...Reviewers will **not** factor the proposed data-sharing plan into the determination of scientific merit or priority score.

[emphasis mine]

<http://grants1.nih.gov/grants/guide/notice-files/NOT-OD-03-032.html>

This must change!

- **Grant review panel**

- Credit proposals that do a good job on this front, note those that don't.

- **Hiring/tenure/promotion committee**

- credit good computational work.

- **Teaching:**

- students must treat computing as rigorously as other parts of the research.

Use your influence to improve the situation

FINAL NIH STATEMENT ON SHARING RESEARCH DATA

...Reviewers will **not** factor the proposed data-sharing plan into the determination of scientific merit or priority score.

[emphasis mine]

<http://grants1.nih.gov/grants/guide/notice-files/NOT-OD-03-032.html>

This must change!

- **Grant review panel**
 - Credit proposals that do a good job on this front, note those that don't.
- **Hiring/tenure/promotion committee**
 - credit good computational work.
- **Teaching:**
 - students must treat computing as rigorously as other parts of the research.



Now accepting
submissions

Editor-in-Chief
Cameron Neylon (UK)

Open Research Computation is an open access journal that publishes articles describing the development, capacities, and uses of software for researchers in any field. The journal also encourages submissions that review or describe developments relating to software based research tools. All software source code published in *Open Research Computation* is made available under an Open Source Initiative compliant license.

Submit your manuscript and benefit from:

- High visibility for articles through unrestricted online access
- No limits on article length, additional files, colour figures or movies
- Immediate open access publication on acceptance
- Expert peer review

www.openresearchcomputation.com