# Reproducible software vs. reproducible research

Fernando Pérez

http://fperez.org

Fernando.Perez@berkeley.edu

Helen Wills Neuroscience Institute, UC Berkeley

AAAS Annual Meeting, 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)**

# Mea culpa: a typical computational publication

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 $\mu \geqslant 0$ 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

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

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

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

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 | $N_{\text{blocks}}$ | $\epsilon$ | $E_2$ |
|---|---|---|---|---|
| 5 | [2,3,4] | 8 | $10^{-3}$ | $1.5 \times 10^{-4}$ |
| 8 | [2,4,5] | 12 | $10^{-6}$ | $1.3 \times 10^{-7}$ |
| 11 | [2,4,5] | 14 | $10^{-9}$ | $1.1 \times 10^{-10}$ |
| 14 | [3,4,5] | 16 | $10^{-12}$ | $4.4 \times 10^{-13}$ |

*Notes*. The order of the basis $p$ is adjusted as a function of the requested precision $\epsilon$. 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 $\ell^2$ norm.

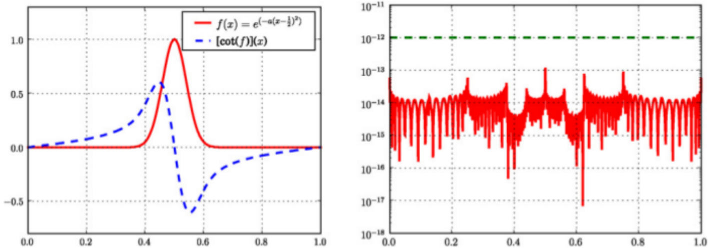
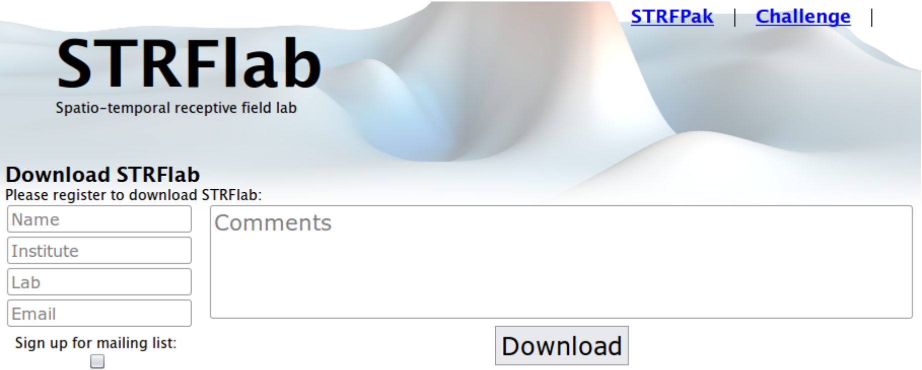



Fig. 3. Results of applying the cotangent kernel to a periodized Gaussian using basis of order $p = 14$ (the last row in Table 1). The pointwise error is shown on the right for a requested accuracy of $\epsilon = 10^{-12}$.

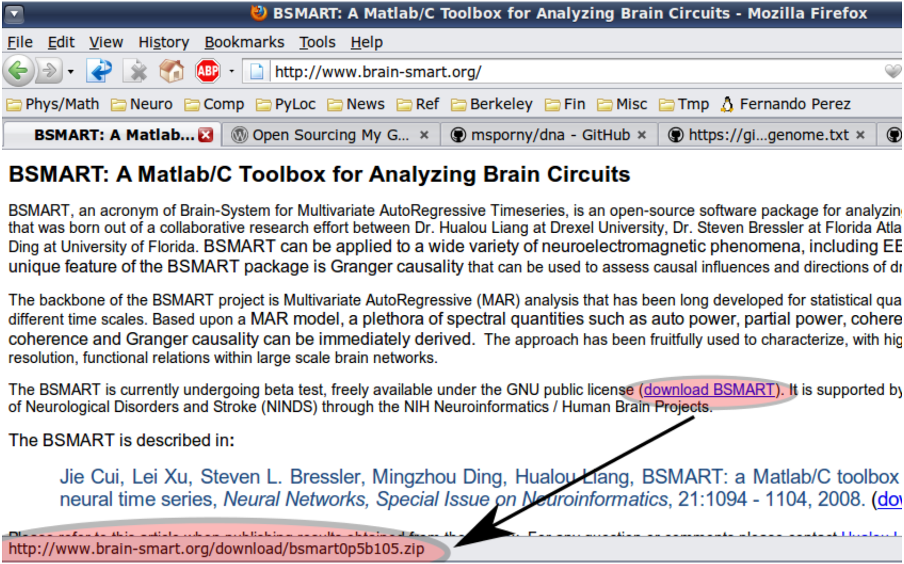
# Chance of reproducing results for third parties?

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

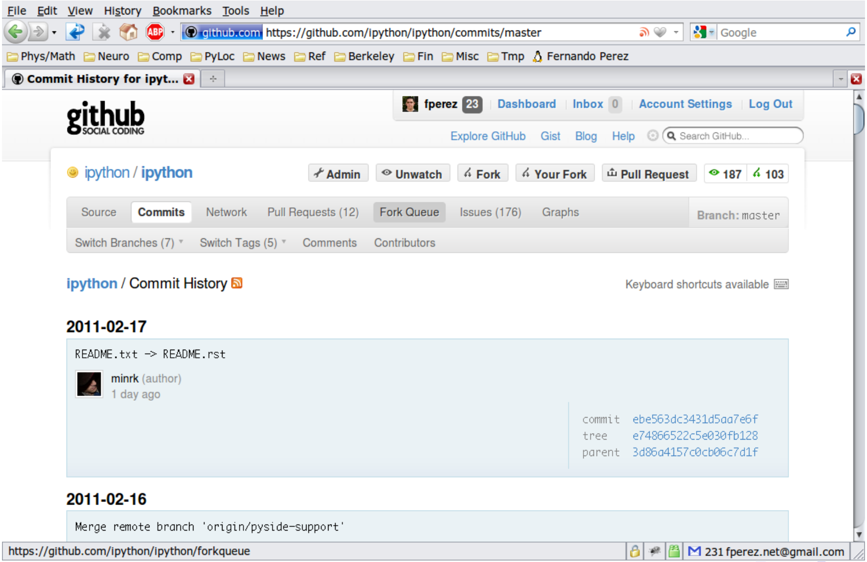
# Registration-wall software



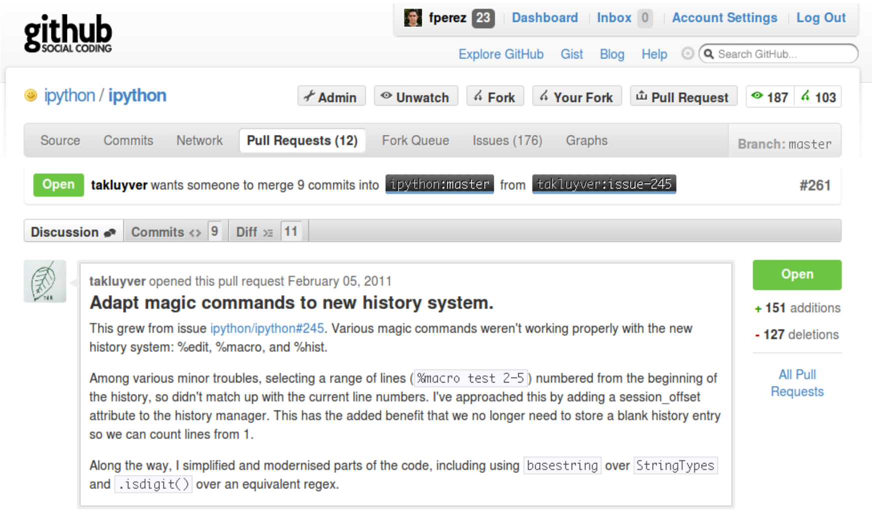
# Zip file/tarball dumps



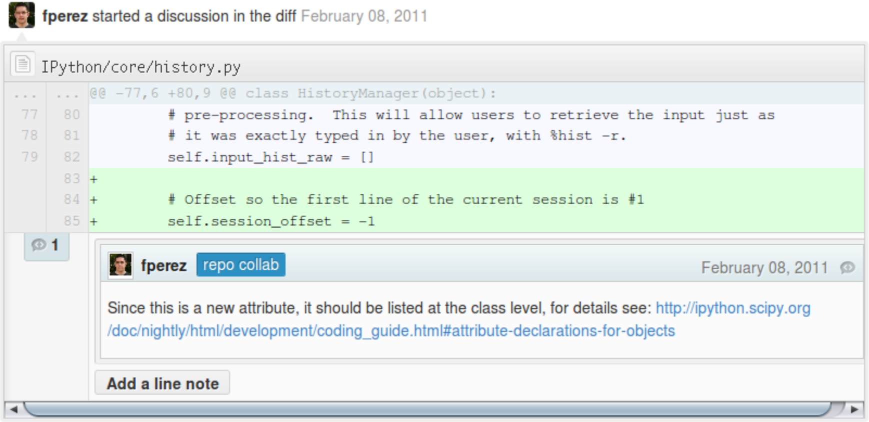
## Public distributed version control: provenance tracking



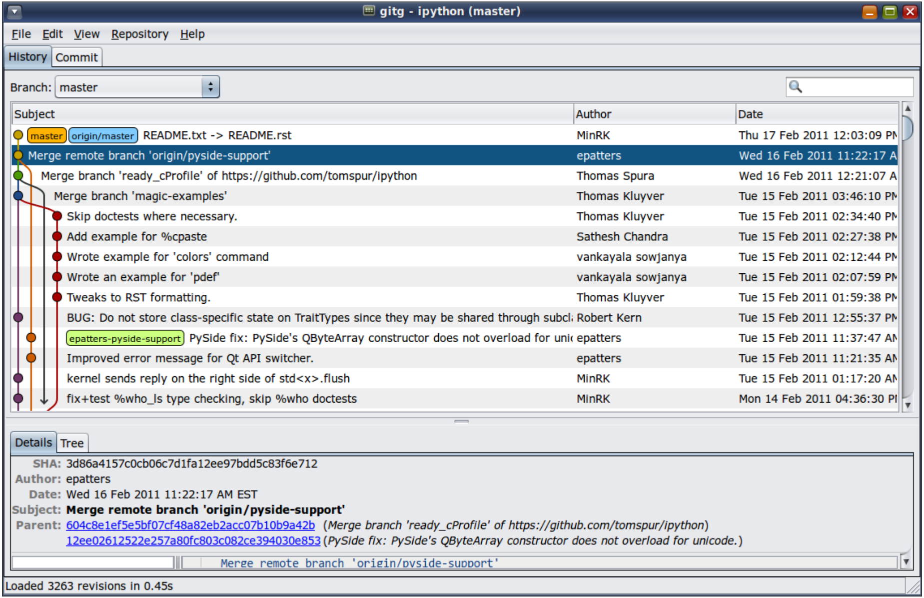
# Pull requests: ongoing peer review



# Pull requests: back and forth discussion

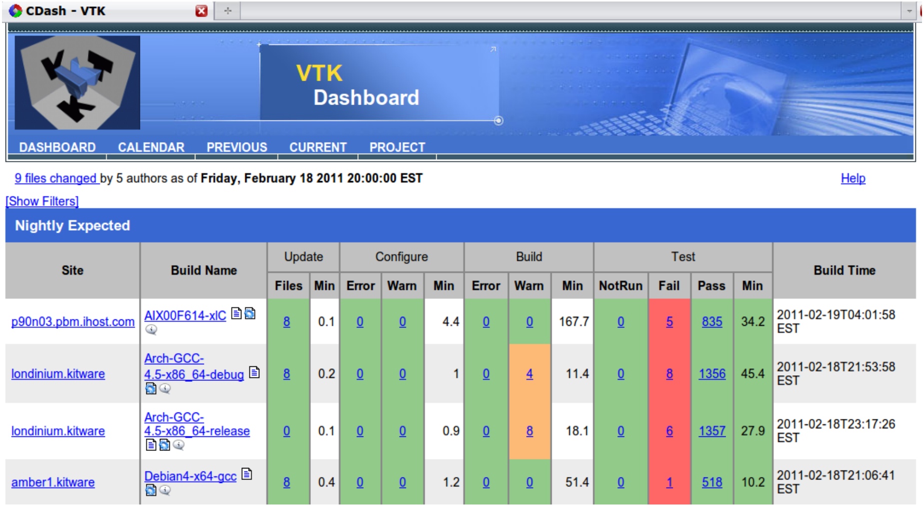


# Branches: exploratory work with control



# Automated tests: validation/reproducibility

## The VTK Build Dashboard: immediate feedback



CDash - VTK

VTK Dashboard

DASHBOARD CALENDAR PREVIOUS CURRENT PROJECT

9 files changed by 5 authors as of **Friday, February 18 2011 20:00:00 EST** Help

[Show Filters]

**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 |
| londinium.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 |
| londinium.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

Search projects

Project Home | Downloads | Wiki | **Issues** | Source

New issue **Search** Open issues **for** Search 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 + 0.111111111111111*(1.00000000000000 + 2.00000000000000*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



## Auditory processing in grasshoppers ¶

Extracting the average time-series from one signal, time-locked to the occurence 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

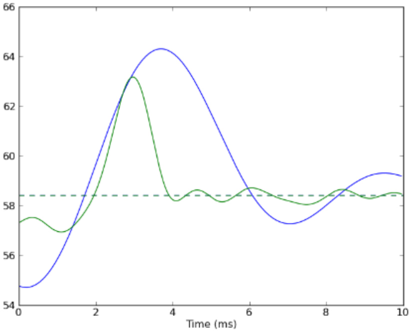
**Previous topic**

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

## 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
- Grant writing
- Everyday research

## Advantages of pervasive DVCS

- Tracking of everyday results. A "time machine" view.
- Distributed backup.
- Explore lines of research/writing.
- Collaborate with colleagues.

## 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
- Grant writing
- Everyday research

## Advantages of pervasive DVCS

- Tracking of everyday results. A "time machine" view.
- Distributed backup.
- Explore lines of research/writing.
- Collaborate with colleagues.

### 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

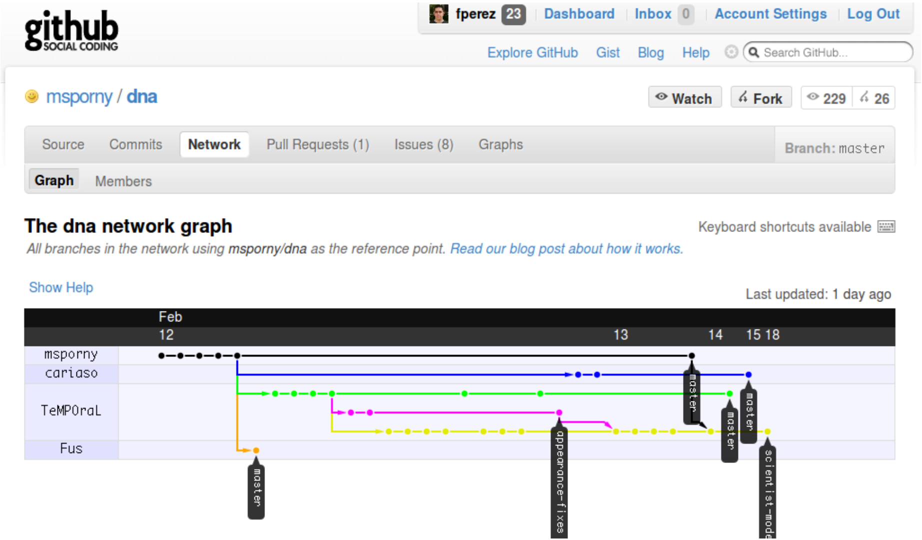
- Paper writing
- Grant writing
- Everyday research

### Advantages of pervasive DVCS

- 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



# 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 authorhsip.
- 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.

- 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]

**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 any other aspect 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]

## 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 any other aspect of the research.