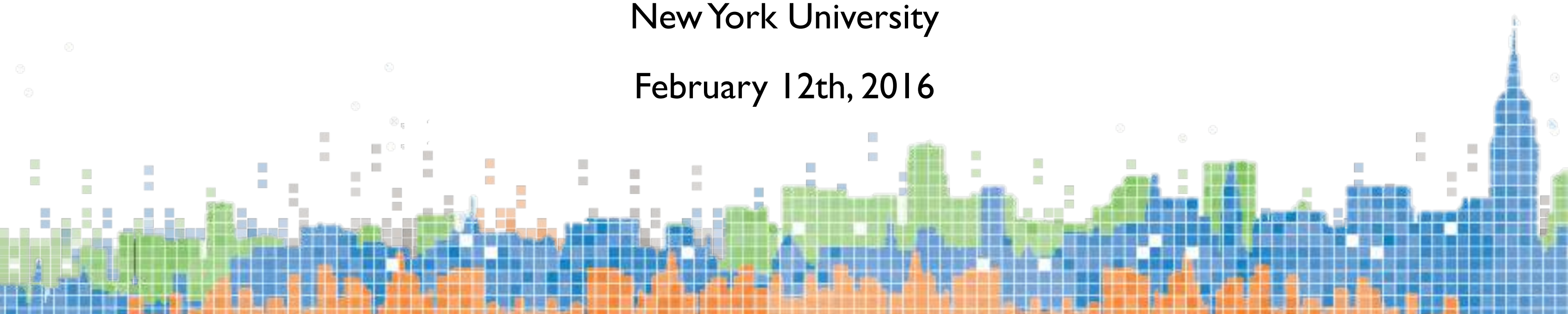
# How Not to Lose Your Code, Your Degree, and Your Future Job

Justin Salamon

Center for Urban Science and Progress

New York University

February 12th, 2016



# How Not To Lose Your Code

(or: How To Write Good Code, Manage It, Share It, and Be Awesome)

## Horror Story 1

Bob: Oh no! Someone broke into our apartment and stole my laptop! All of my code is gone…

Alice: I thought you kept a backup of your code on an external hard drive, no?

Bob: They took that too…

## Horror Story 2

Bob: dammit, I've made some changes to my code and now it doesn't work anymore

Alice: that's ok Bob. Just revert to a version of the code you know worked.

Bob: ehm… version?

## Horror Story 3

Bob: I was collaborating with Rob on a project, and we were emailing each other the latest version of the code. Over spring break we both worked separately, and now we don't know how to merge our work!

Alice: that's ok Bob. Just use Git to compare the two repositories, identify conflicting sections of the code and help you merge them back into one.

Bob: ehm… Git?

# Horror Story 4
(true!)

Ex-master student: Hey Justin, I was reading your paper from a few years ago , and I was wondering how exactly you normalize the data in Section 5?

Justin: Oh god… let me find my PhD backup drive…

[finds drive]

Justin: Oh god… let me find the code used for that paper on the drive…

[finds code]

Justin: Oh god… let me figure out which version of the code w d when the paper was written…

[after much suffering, figures out which version of the code
and how normalization was applied. Fail averted. Lucki

[kicks himself for not using version control at the tin

## Version Control

Software to help keep track of changes made to files:

- Tracks the **history** of your work
- Helps you **collaborate** with others
- Helps keep an **online backup** of your code

## Keeping Track of History

- How do you get back to that **working version** you had yesterday?
- How do you get from "it's not working" to **understanding what went wrong**?
- How would you **repeat** the experiments from that paper (or project) you worked on last year?

## Collaborating

With yourself:

- Need to run same code on laptop and on CUSP server:
  - How do you get the code onto both?
  - How do **verify** that you have the same code on both?

With others:

- Working on a project with classmates:
  - How do you find out when they changes something?
  - How do you merge your changes without making a mess?
  - How can you find out which of you introduced a bug, and when?

## Version Control

A version control system:

- Records you files' **history**
- Shows the **differences** between versions
- Handles **sync** between copies on different computes and **the cloud**

## How is that Different from Dropbox?

Guarantees consistency:

- Code needs to be **exactly as written** across all files
- Files changed together can be **updated together**
- Changes to a file by different people must be **merged correctly**

Keeps records:

- Publish and replicate history reliably
- Interrogate past **changes** and find out what version you are looking at

## Version Control Evolution

- Revision Control System (RCS): 1982
  - Each file versioned independently, manual sync ☹
- Concurrent Versions System (CVS): 1986, 1990
  - Multiple file versioning ☺
  - Can only change latest version, clumsy networking, poor support for binary files ☹
- Subversion (SVN): 2000
  - Similar to CVS but with many improvements ☺
  - Versioning done on server-side: local dev is tricky ☹
  - Single point of failure (the server) ☹

## How Not To Lose Your Code : Git

git

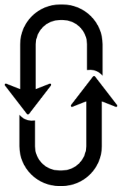
# Git

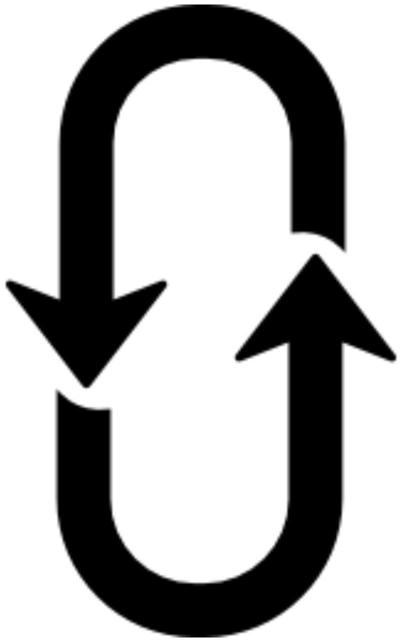
[Torvalds, 2005]

- Features:
  - **Distributed** version control system (DVCS)
  - Does not require a centralized server
    - But you can still have one, if you want
  - Easy local incremental development
  - No single point of failure
    - Every developer has local copy with full history of the repository
- Drawbacks:
  - A bit of a learning curve…
- Other DVCs
  - Mercurial (`hg`)
  - Bazaar (`bzr`)

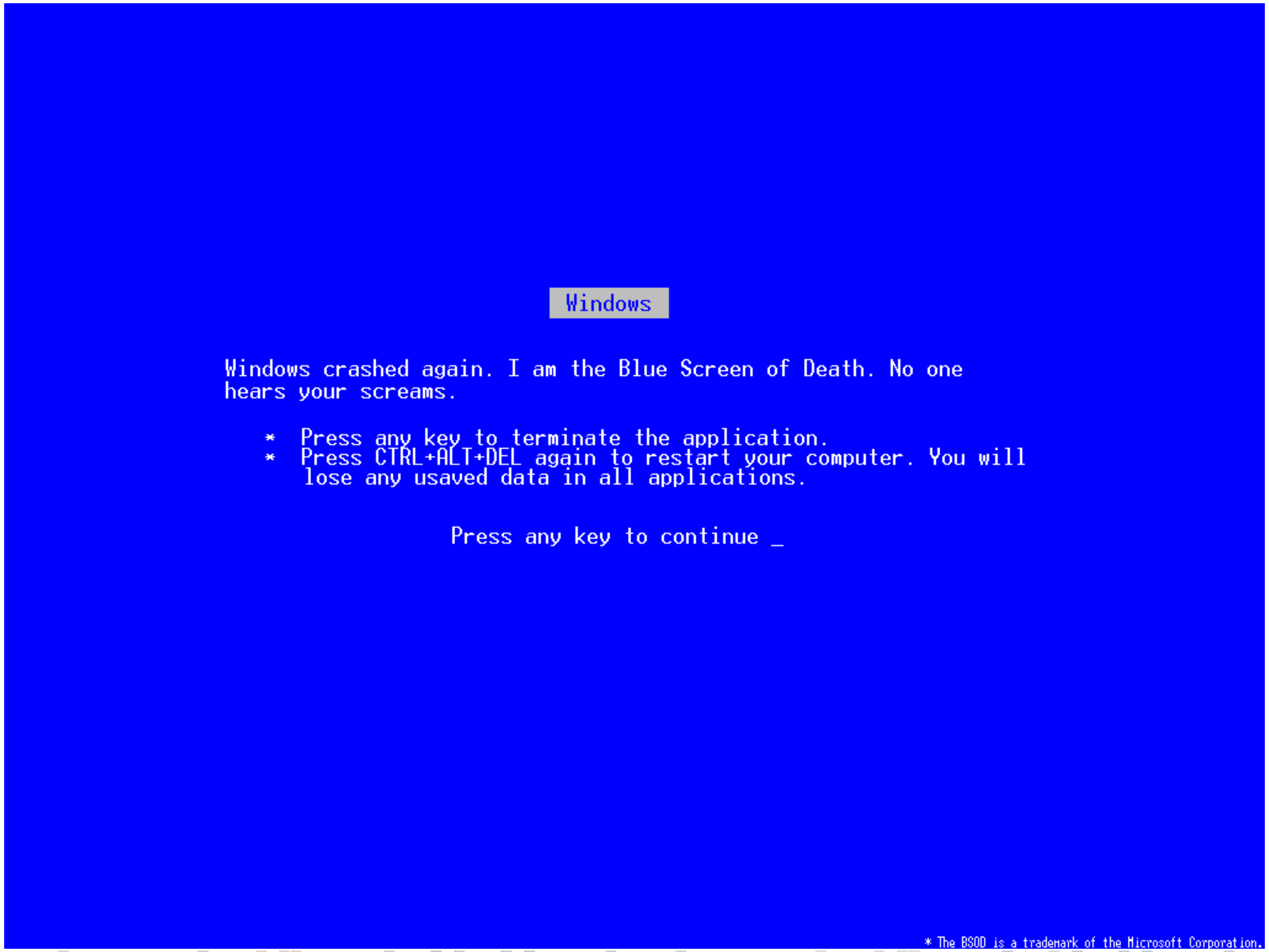
## Using Git: client-server

| | | |
|---|---|---|
| 1. | `git clone` | Make local copy of the repo |
| 2. | (edit files) | |
| 3. | `git commit` | Register your changes locally |
| 4. | `git push` | Share changes upstream |
| 5. | `git pull` | Get updates from upstream |





## Demo



Windows

Windows crashed again. I am the Blue Screen of Death. No one hears your screams.

* Press any key to terminate the application.
* Press CTRL+ALT+DEL again to restart your computer. You will lose any usaved data in all applications.

Press any key to continue _

* The BSOD is a trademark of the Microsoft Corporation.

## Advanced Usage: Tags

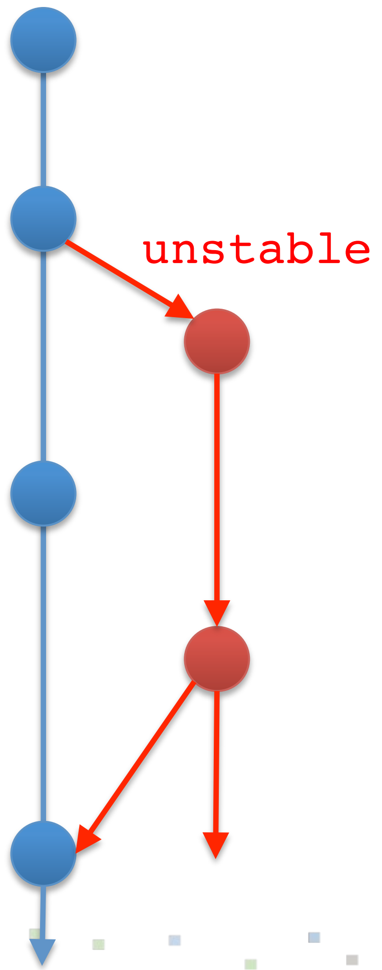
- Some revisions are special:
  - Initial paper submission
  - Camera ready submission
  - Public software releases
- Tagging links **semantic versions** to **revisions**
- Example:
  - `git tag —a v1.0`
  - `git push origin --tags`

## Advanced Usage: Branches

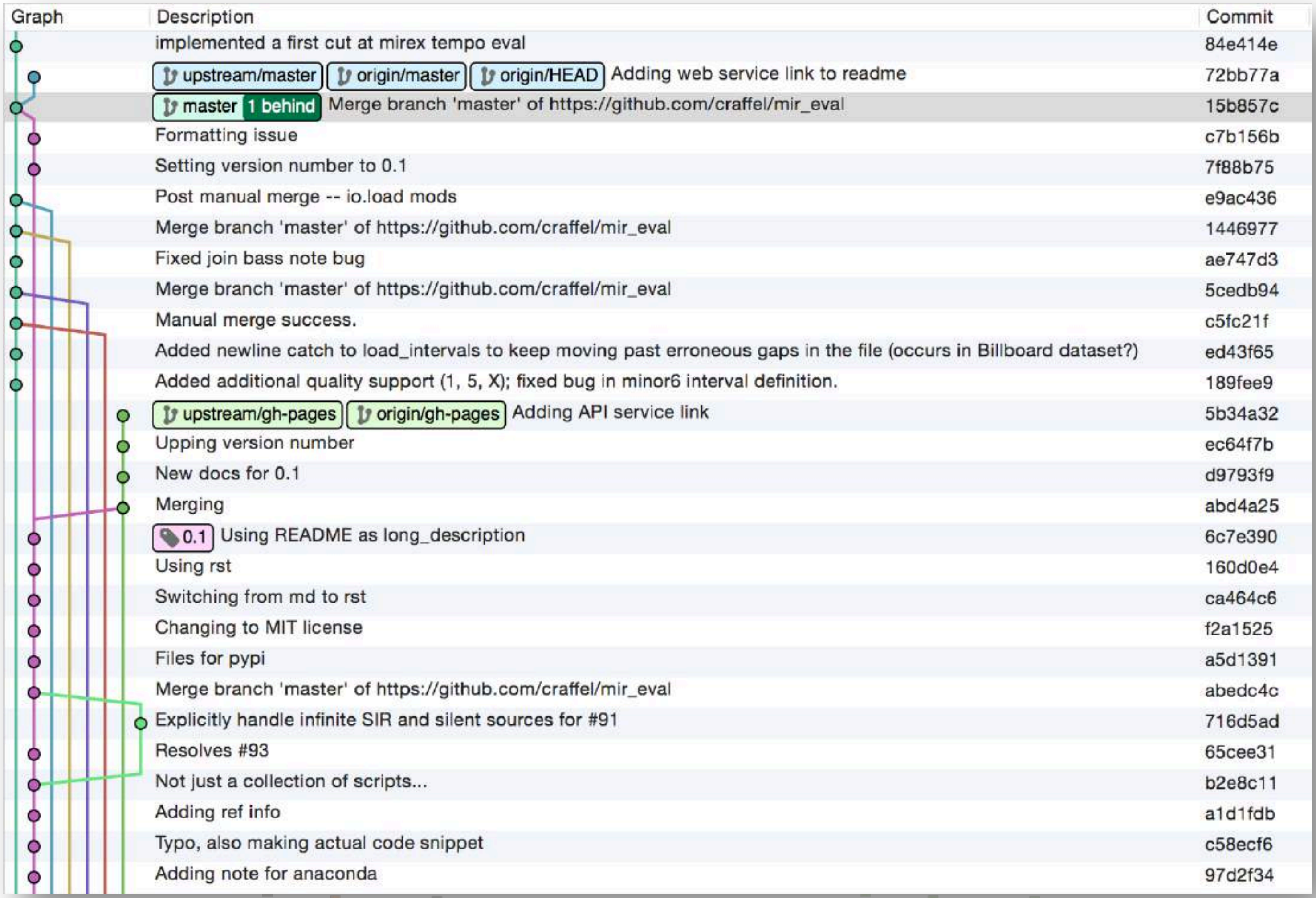
- What if you want to develop new features, but retain version control on a stable codebase?
- Working in a **branch** of the source tree
- Merge back when you're ready
- Especially useful for collaborations

## Advanced Usage: Branches

- Example: create a new branch
  - `git checkout –b unstable`
  - `(edits, commits, pushes)`
- Switch to master, bug fix, switch back
  - `git checkout master`
  - `(edits, commits, pushes)`
  - `git checkout unstable`
- Merge unstable back into master
  - `git checkout master`
  - `git merge unstable`

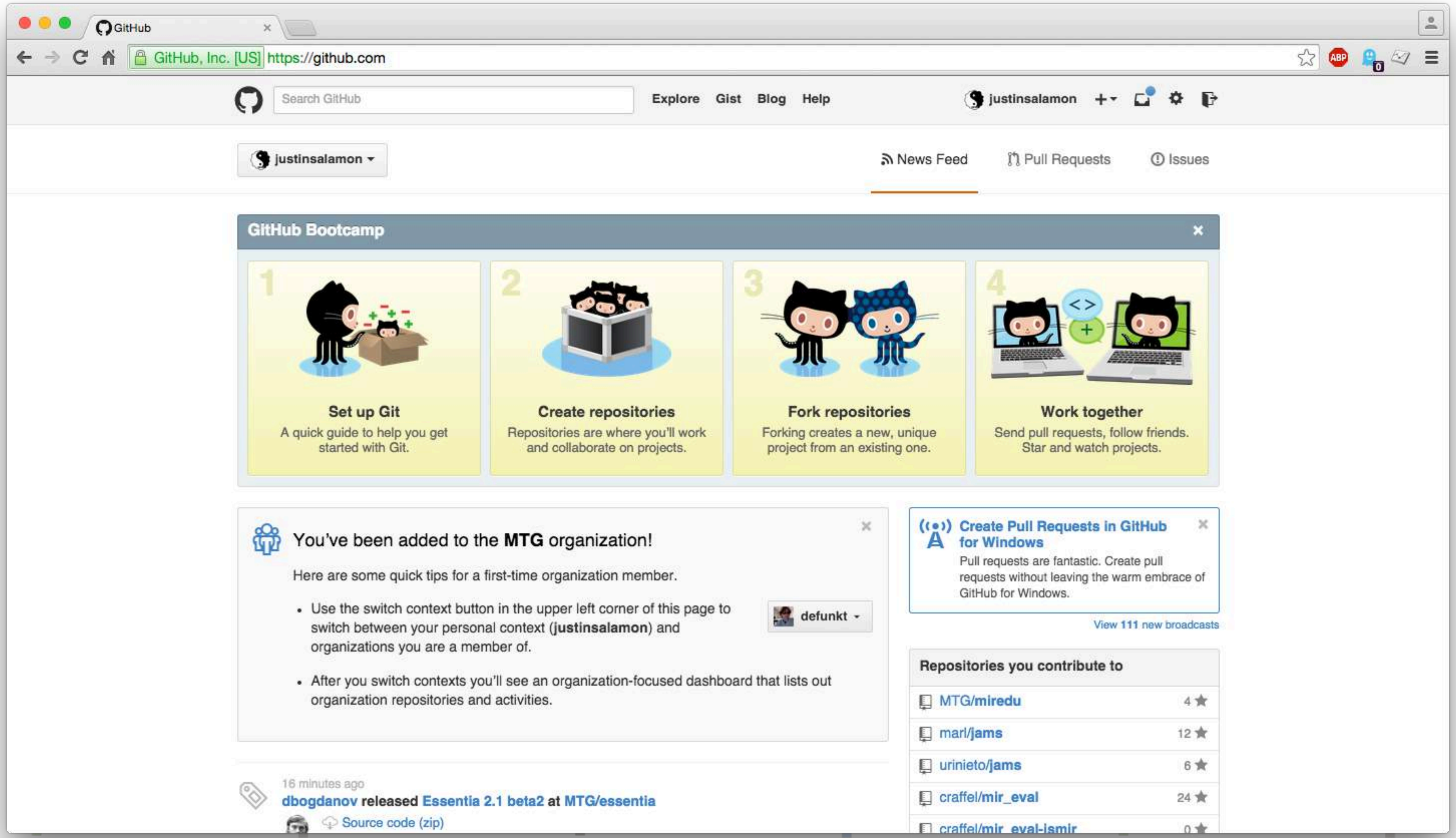


# Advanced Usage: Branches



| Graph | Description | Commit |
|---|---|---|
| | implemented a first cut at mirex tempo eval | 84e414e |
| | upstream/master origin/master origin/HEAD Adding web service link to readme | 72bb77a |
| | master 1 behind Merge branch 'master' of https://github.com/craffel/mir_eval | 15b857c |
| | Formatting issue | c7b156b |
| | Setting version number to 0.1 | 7f88b75 |
| | Post manual merge -- io.load mods | e9ac436 |
| | Merge branch 'master' of https://github.com/craffel/mir_eval | 1446977 |
| | Fixed join bass note bug | ae747d3 |
| | Merge branch 'master' of https://github.com/craffel/mir_eval | 5cedb94 |
| | Manual merge success. | c5fc21f |
| | Added newline catch to load_intervals to keep moving past erroneous gaps in the file (occurs in Billboard dataset?) | ed43f65 |
| | Added additional quality support (1, 5, X); fixed bug in minor6 interval definition. | 189fee9 |
| | upstream/gh-pages origin/gh-pages Adding API service link | 5b34a32 |
| | Upping version number | ec64f7b |
| | New docs for 0.1 | d9793f9 |
| | Merging | abd4a25 |
| | 0.1 Using README as long_description | 6c7e390 |
| | Using rst | 160d0e4 |
| | Switching from md to rst | ca464c6 |
| | Changing to MIT license | f2a1525 |
| | Files for pypi | a5d1391 |
| | Merge branch 'master' of https://github.com/craffel/mir_eval | abedc4c |
| | Explicitly handle infinite SIR and silent sources for #91 | 716d5ad |
| | Resolves #93 | 65cee31 |
| | Not just a collection of scripts... | b2e8c11 |
| | Adding ref info | a1d1fdb |
| | Typo, also making actual code snippet | c58ecf6 |
| | Adding note for anaconda | 97d2f34 |

## Hosting Git Online



**Set up Git**
A quick guide to help you get started with Git.

**Create repositories**
Repositories are where you'll work and collaborate on projects.

**Fork repositories**
Forking creates a new, unique project from an existing one.

**Work together**
Send pull requests, follow friends. Star and watch projects.

You've been added to the **MTG** organization!

Here are some quick tips for a first-time organization member.

- Use the switch context button in the upper left corner of this page to switch between your personal context (**justinsalamon**) and organizations you are a member of.
- After you switch contexts you'll see an organization-focused dashboard that lists out organization repositories and activities.

defunkt

16 minutes ago
**dbogdanov** released **Essentia 2.1 beta2** at **MTG/essentia**
Source code (zip)

**Create Pull Requests in GitHub for Windows**
Pull requests are fantastic. Create pull requests without leaving the warm embrace of GitHub for Windows.

View **111** new broadcasts

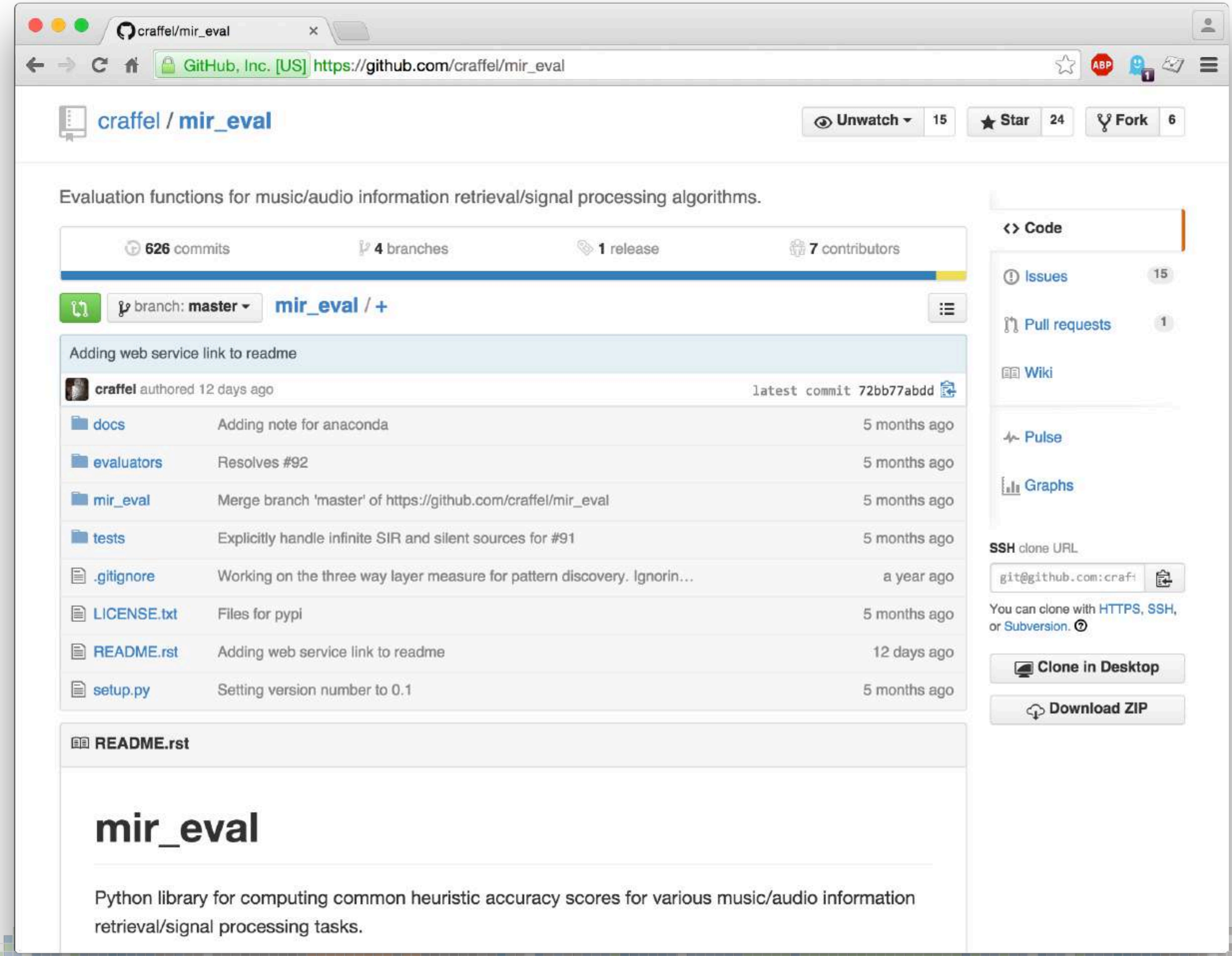
**Repositories you contribute to**

| | |
|---|---|
| MTG/miredu | 4 ★ |
| marl/jams | 12 ★ |
| urinieto/jams | 6 ★ |
| craffel/mir_eval | 24 ★ |
| craffel/mir_eval-ismir | 0 ★ |

## GitHub

- Free hosting for open source projects
  - Free organization accounts for academics
- Social network integration
- **Extra usability tools**
  - User management
  - Pull requests
  - Issue tracking, comments, wiki
  - Release management
  - Webhooks & services!
- **Portfolio for potential employers!**
  - More on this later

# How Not To Lose Your Code : GitHub



craffel/mir_eval

GitHub, Inc. [US] https://github.com/craffel/mir_eval

craffel / mir_eval

Unwatch 15 · Star 24 · Fork 6

Evaluation functions for music/audio information retrieval/signal processing algorithms.

626 commits · 4 branches · 1 release · 7 contributors

branch: master · mir_eval / +

Adding web service link to readme

craffel authored 12 days ago · latest commit 72bb77abdd

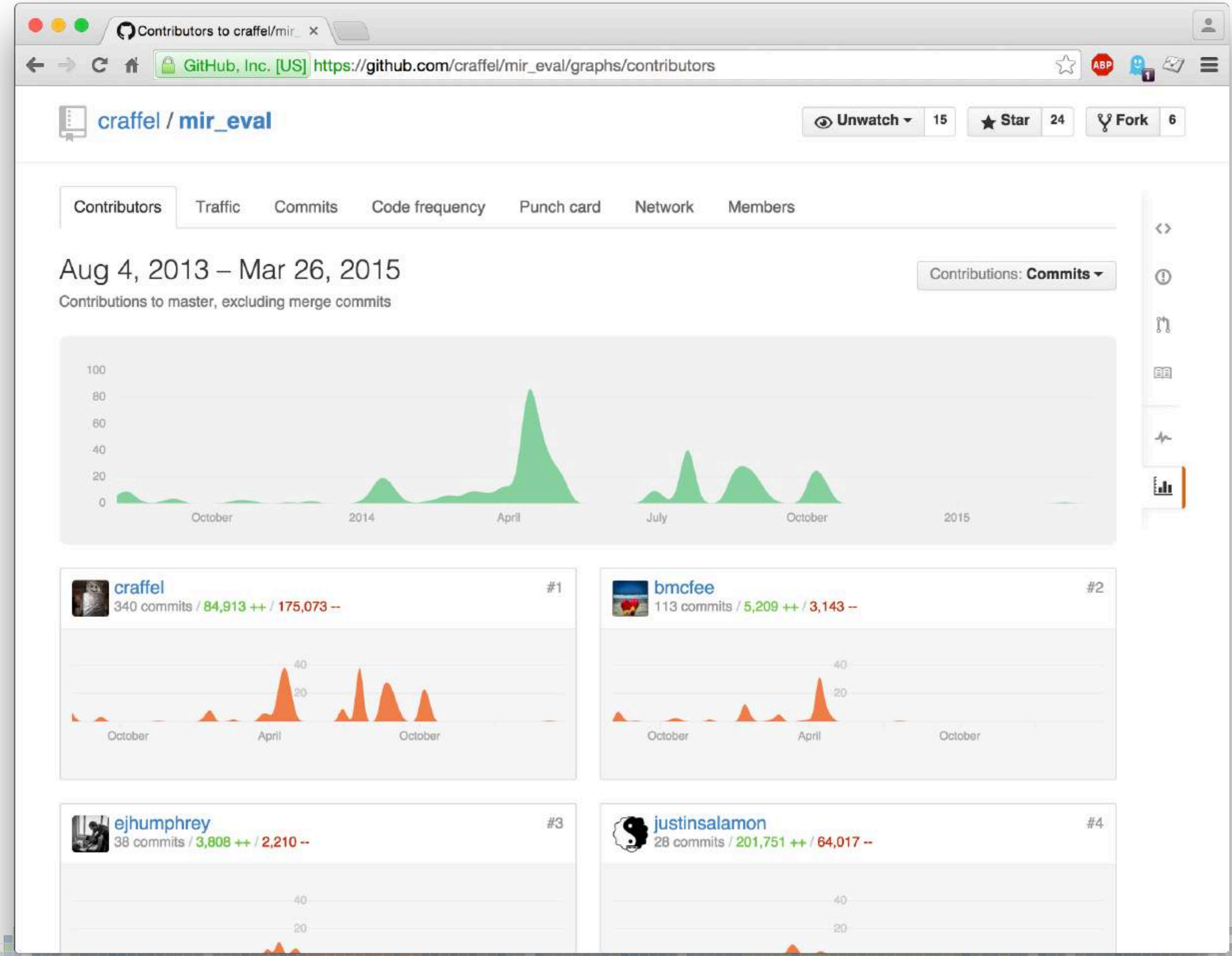
| | | |
|---|---|---|
| docs | Adding note for anaconda | 5 months ago |
| evaluators | Resolves #92 | 5 months ago |
| mir_eval | Merge branch 'master' of https://github.com/craffel/mir_eval | 5 months ago |
| tests | Explicitly handle infinite SIR and silent sources for #91 | 5 months ago |
| .gitignore | Working on the three way layer measure for pattern discovery. Ignorin… | a year ago |
| LICENSE.txt | Files for pypi | 5 months ago |
| README.rst | Adding web service link to readme | 12 days ago |
| setup.py | Setting version number to 0.1 | 5 months ago |

README.rst

mir_eval

Python library for computing common heuristic accuracy scores for various music/audio information retrieval/signal processing tasks.

# How Not To Lose Your Code : GitHub


Contributors to craffel/mir_

GitHub, Inc. [US] https://github.com/craffel/mir_eval/graphs/contributors

craffel / mir_eval

Aug 4, 2013 – Mar 26, 2015

Contributions to master, excluding merge commits

Contributions: Commits

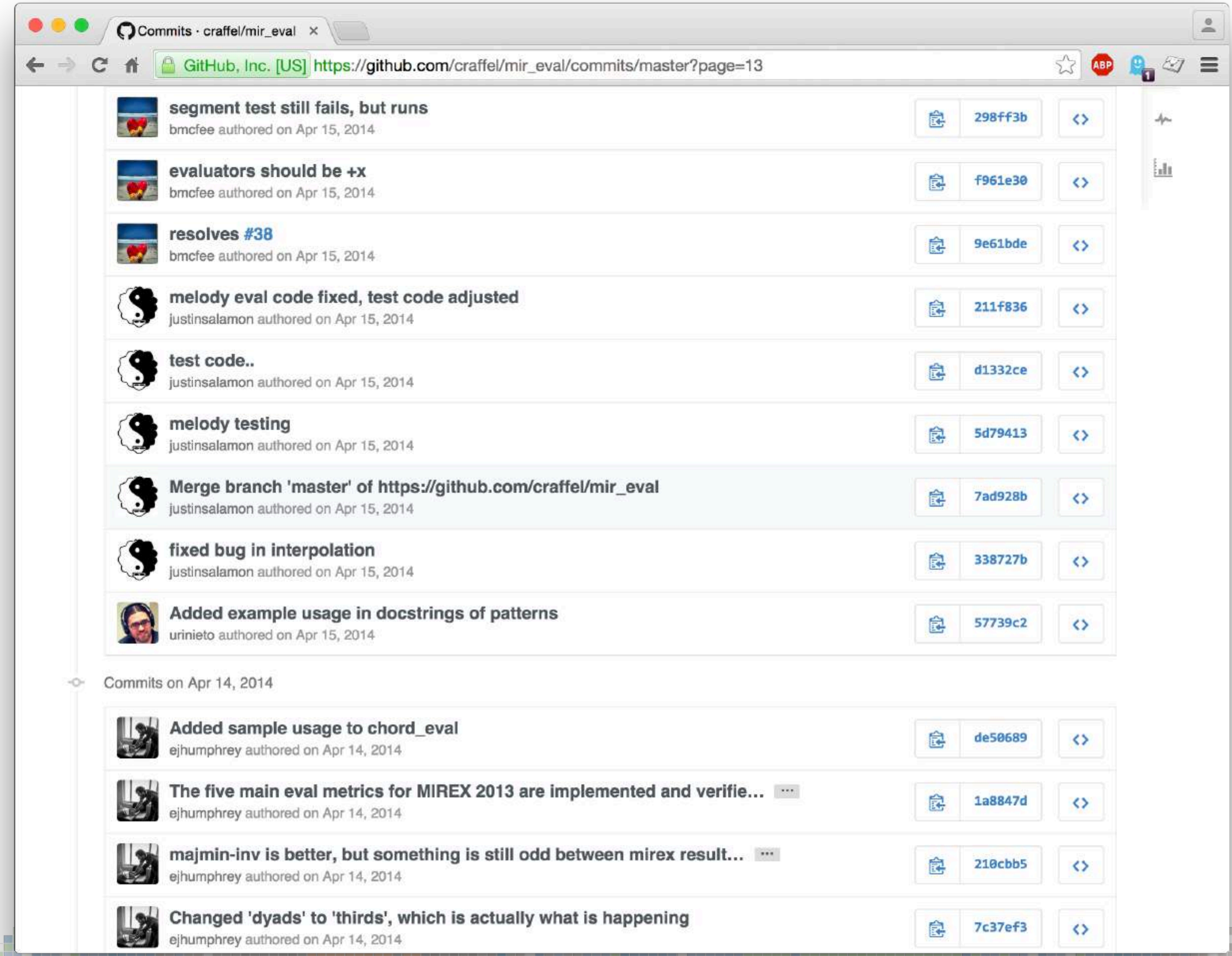
craffel #1
340 commits / 84,913 ++ / 175,073 --

bmcfee #2
113 commits / 5,209 ++ / 3,143 --

ejhumphrey #3
38 commits / 3,808 ++ / 2,210 --

justinsalamon #4
28 commits / 201,751 ++ / 64,017 --

## How Not To Lose Your Code : GitHub


Commits · craffel/mir_eval
GitHub, Inc. [US] https://github.com/craffel/mir_eval/commits/master?page=13

segment test still fails, but runs
bmcfee authored on Apr 15, 2014 — 298ff3b

evaluators should be +x
bmcfee authored on Apr 15, 2014 — f961e30

resolves #38
bmcfee authored on Apr 15, 2014 — 9e61bde

melody eval code fixed, test code adjusted
justinsalamon authored on Apr 15, 2014 — 211f836

test code..
justinsalamon authored on Apr 15, 2014 — d1332ce

melody testing
justinsalamon authored on Apr 15, 2014 — 5d79413

Merge branch 'master' of https://github.com/craffel/mir_eval
justinsalamon authored on Apr 15, 2014 — 7ad928b

fixed bug in interpolation
justinsalamon authored on Apr 15, 2014 — 338727b

Added example usage in docstrings of patterns
urinieto authored on Apr 15, 2014 — 57739c2

Commits on Apr 14, 2014

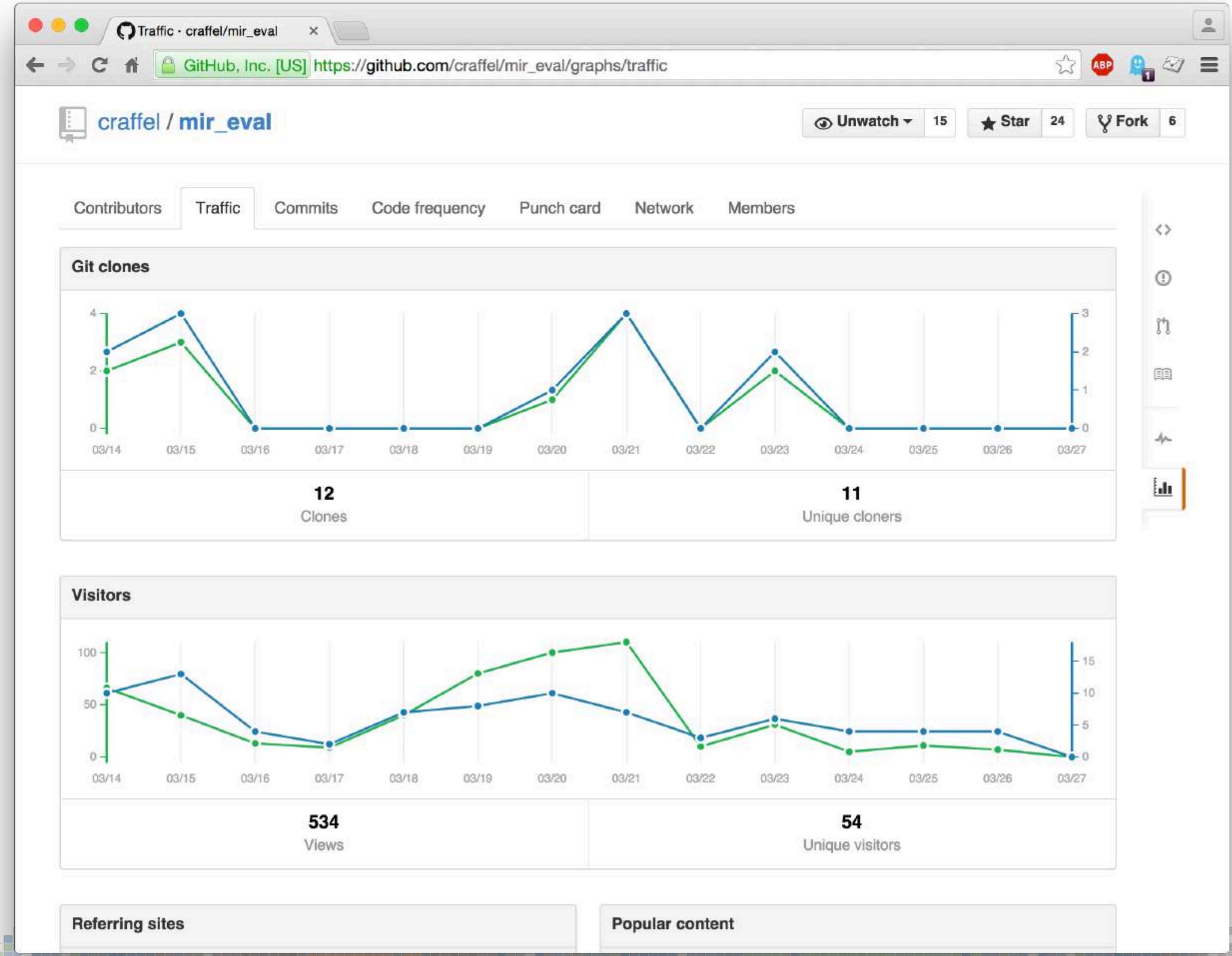
Added sample usage to chord_eval
ejhumphrey authored on Apr 14, 2014 — de50689

The five main eval metrics for MIREX 2013 are implemented and verifie… …
ejhumphrey authored on Apr 14, 2014 — 1a8847d

majmin-inv is better, but something is still odd between mirex result… …
ejhumphrey authored on Apr 14, 2014 — 210cbb5

Changed 'dyads' to 'thirds', which is actually what is happening
ejhumphrey authored on Apr 14, 2014 — 7c37ef3

# How Not To Lose Your Code : GitHub



Git clones

12 Clones

11 Unique cloners

Visitors

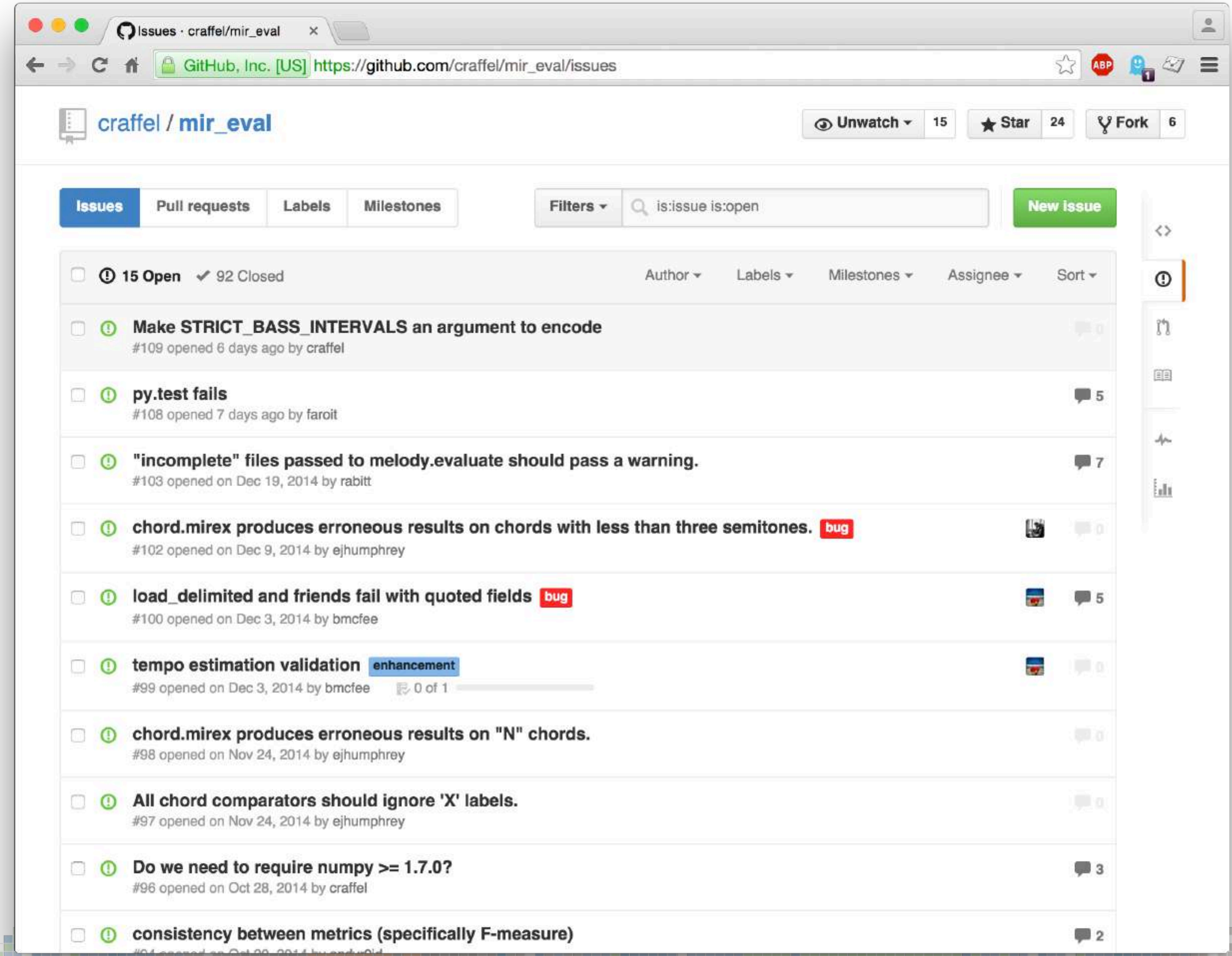
534 Views

54 Unique visitors

Referring sites

Popular content

## How Not To Lose Your Code : GitHub



Make STRICT_BASS_INTERVALS an argument to encode
#109 opened 6 days ago by craffel

py.test fails — 5
#108 opened 7 days ago by faroit

"incomplete" files passed to melody.evaluate should pass a warning. — 7
#103 opened on Dec 19, 2014 by rabitt

chord.mirex produces erroneous results on chords with less than three semitones. bug
#102 opened on Dec 9, 2014 by ejhumphrey

load_delimited and friends fail with quoted fields bug — 5
#100 opened on Dec 3, 2014 by bmcfee

tempo estimation validation enhancement
#99 opened on Dec 3, 2014 by bmcfee 0 of 1

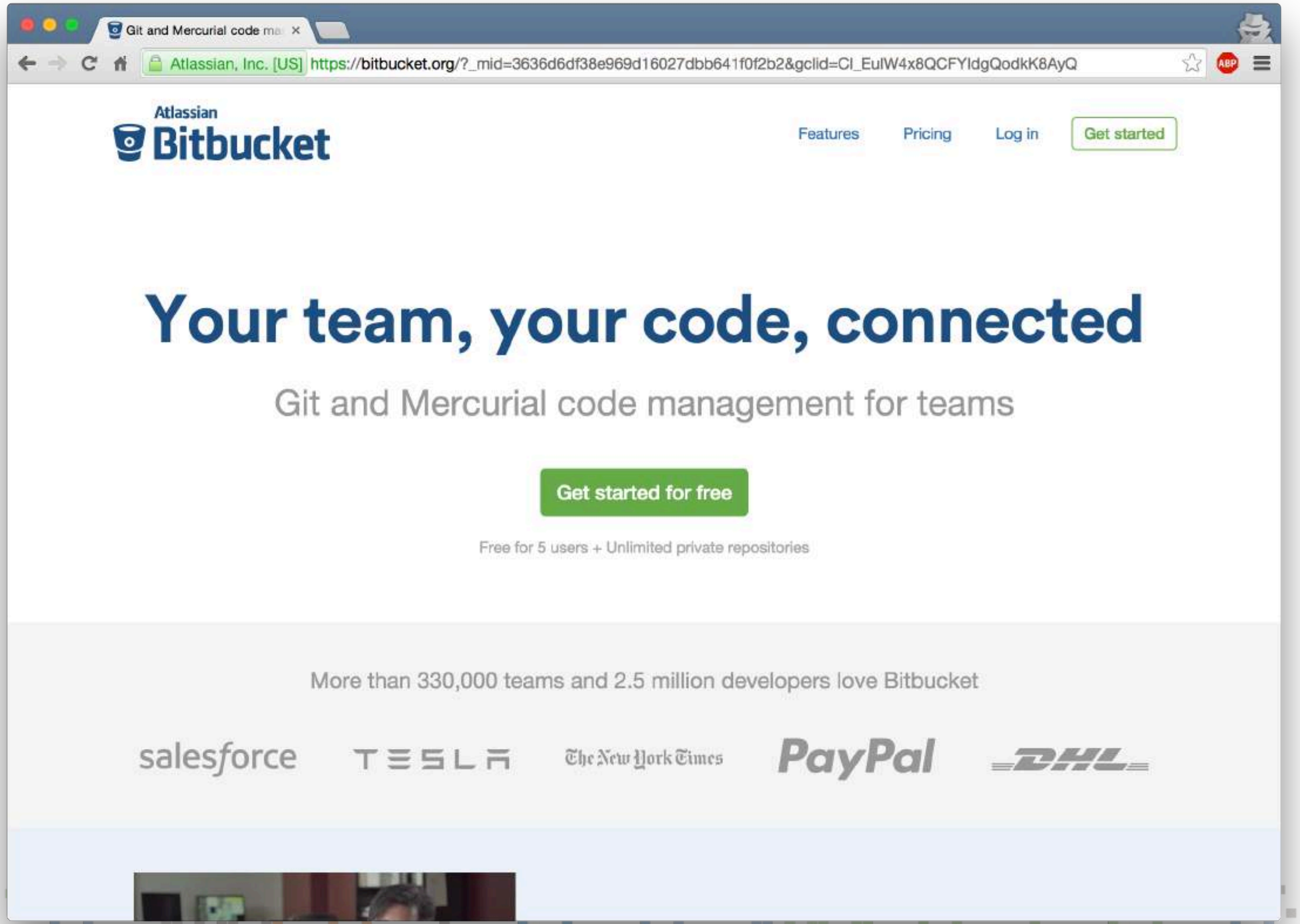
chord.mirex produces erroneous results on "N" chords.
#98 opened on Nov 24, 2014 by ejhumphrey

All chord comparators should ignore 'X' labels.
#97 opened on Nov 24, 2014 by ejhumphrey

Do we need to require numpy >= 1.7.0? — 3
#96 opened on Oct 28, 2014 by craffel

consistency between metrics (specifically F-measure) — 2

## What About Private Repositories?


Git and Mercurial code ma
Atlassian, Inc. [US] https://bitbucket.org/?_mid=3636d6df38e969d16027dbb641f0f2b2&gclid=CI_EulW4x8QCFYldgQodkK8AyQ
Atlassian Bitbucket
Features
Pricing
Log in
Get started
Get started for free

Your team, your code, connected

Git and Mercurial code management for teams

Free for 5 users + Unlimited private repositories

More than 330,000 teams and 2.5 million developers love Bitbucket

salesforce
TESLA
The New York Times
PayPal
DHL

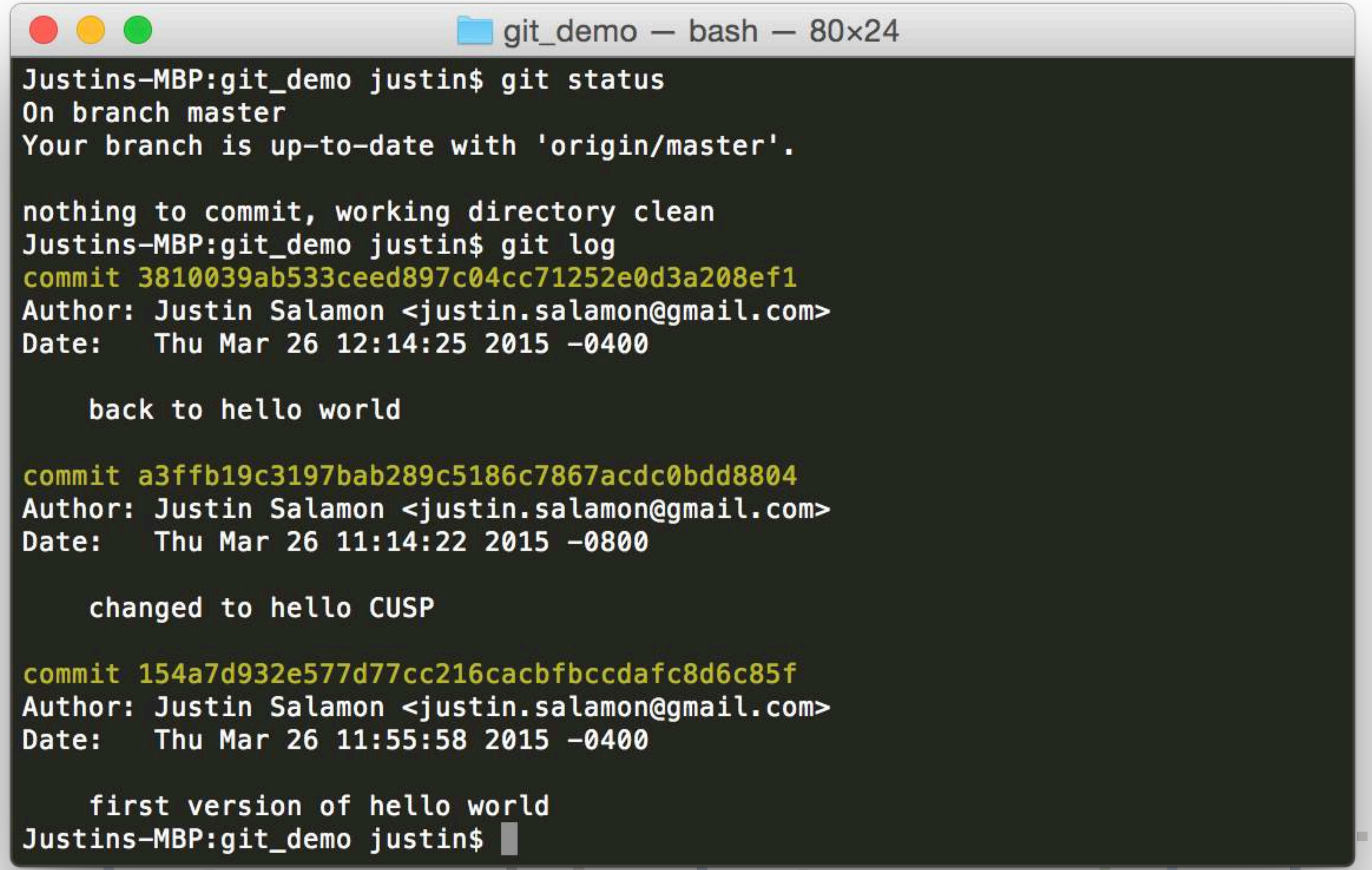
## What About Sensitive Code/Data?

CUSP Data Facility can host Git repo's!

## What if I'm not sure?

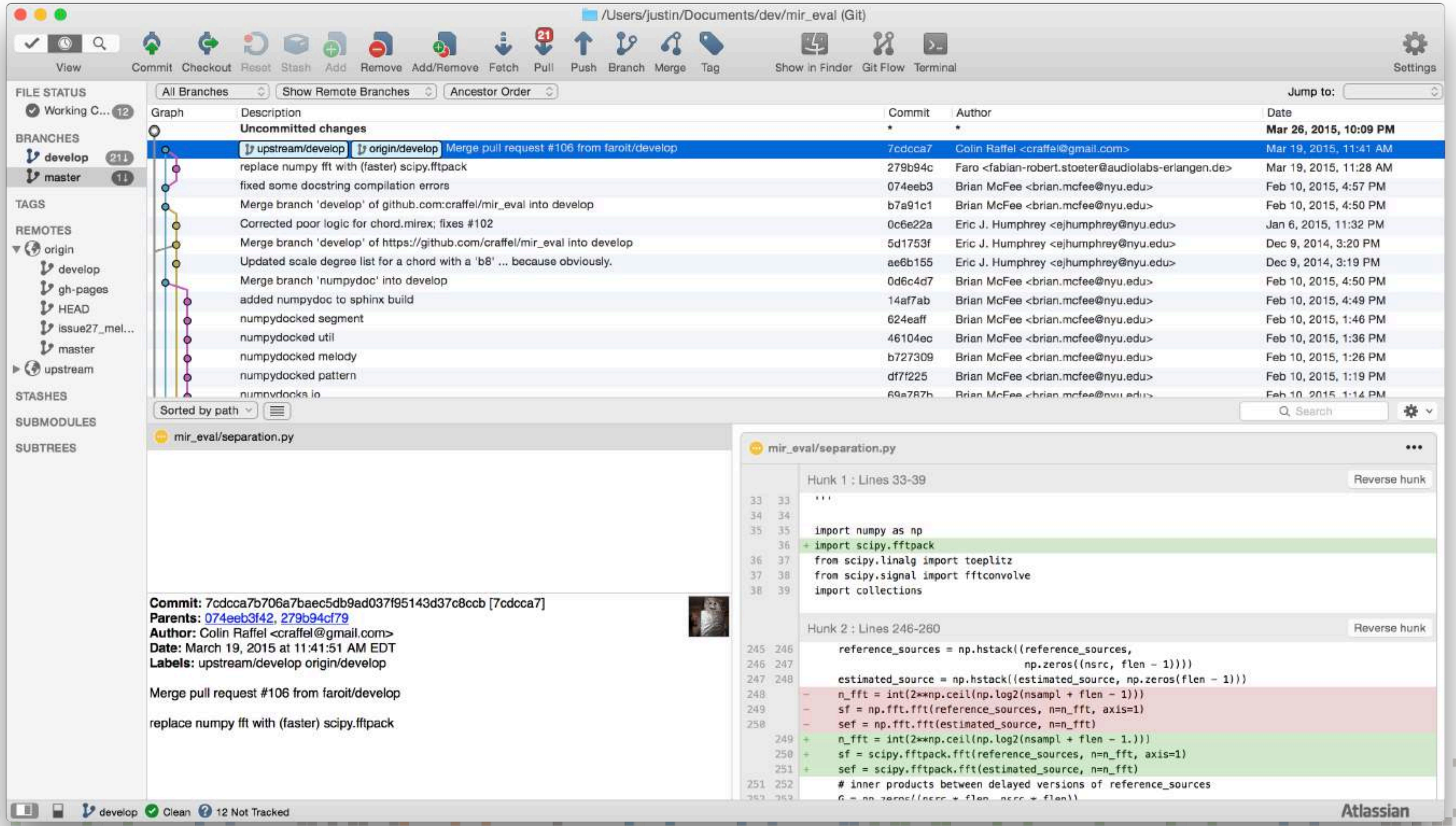
CUSP Data Facility

## Git In Practice



```
Justins-MBP:git_demo justin$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean
Justins-MBP:git_demo justin$ git log
commit 3810039ab533ceed897c04cc71252e0d3a208ef1
Author: Justin Salamon <justin.salamon@gmail.com>
Date:   Thu Mar 26 12:14:25 2015 -0400

    back to hello world

commit a3ffb19c3197bab289c5186c7867acdc0bdd8804
Author: Justin Salamon <justin.salamon@gmail.com>
Date:   Thu Mar 26 11:14:22 2015 -0800

    changed to hello CUSP

commit 154a7d932e577d77cc216cacbfbccdafc8d6c85f
Author: Justin Salamon <justin.salamon@gmail.com>
Date:   Thu Mar 26 11:55:58 2015 -0400

    first version of hello world
Justins-MBP:git_demo justin$
```

## What if I’m More of a Visual Type?



| Description | Commit | Author | Date |
|---|---|---|---|
| **Uncommitted changes** | * | * | **Mar 26, 2015, 10:09 PM** |
| upstream/develop origin/develop Merge pull request #106 from faroit/develop | 7cdcca7 | Colin Raffel <craffel@gmail.com> | Mar 19, 2015, 11:41 AM |
| replace numpy fft with (faster) scipy.fftpack | 279b94c | Faro <fabian-robert.stoeter@audiolabs-erlangen.de> | Mar 19, 2015, 11:28 AM |
| fixed some docstring compilation errors | 074eeb3 | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 4:57 PM |
| Merge branch 'develop' of github.com:craffel/mir_eval into develop | b7a91c1 | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 4:50 PM |
| Corrected poor logic for chord.mirex; fixes #102 | 0c6e22a | Eric J. Humphrey <ejhumphrey@nyu.edu> | Jan 6, 2015, 11:32 PM |
| Merge branch 'develop' of https://github.com/craffel/mir_eval into develop | 5d1753f | Eric J. Humphrey <ejhumphrey@nyu.edu> | Dec 9, 2014, 3:20 PM |
| Updated scale degree list for a chord with a 'b8' ... because obviously. | ae6b155 | Eric J. Humphrey <ejhumphrey@nyu.edu> | Dec 9, 2014, 3:19 PM |
| Merge branch 'numpydoc' into develop | 0d6c4d7 | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 4:50 PM |
| added numpydoc to sphinx build | 14af7ab | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 4:49 PM |
| numpydocked segment | 624eaff | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 1:46 PM |
| numpydocked util | 46104ec | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 1:36 PM |
| numpydocked melody | b727309 | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 1:26 PM |
| numpydocked pattern | df7f225 | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 1:19 PM |
| numpydocks.io | 69a787b | Brian McFee <brian.mcfee@nyu.edu> | Feb 10, 2015, 1:14 PM |

mir_eval/separation.py

**Commit:** 7cdcca7b706a7baec5db9ad037f95143d37c8ccb [7cdcca7]
**Parents:** 074eeb3f42, 279b94cf79
**Author:** Colin Raffel <craffel@gmail.com>
**Date:** March 19, 2015 at 11:41:51 AM EDT
**Labels:** upstream/develop origin/develop

Merge pull request #106 from faroit/develop

replace numpy fft with (faster) scipy.fftpack

mir_eval/separation.py

Hunk 1 : Lines 33-39

```
  '''

  import numpy as np
+ import scipy.fftpack
  from scipy.linalg import toeplitz
  from scipy.signal import fftconvolve
  import collections
```

Hunk 2 : Lines 246-260

```
      reference_sources = np.hstack((reference_sources,
                                     np.zeros((nsrc, flen - 1))))
      estimated_source = np.hstack((estimated_source, np.zeros(flen - 1)))
-     n_fft = int(2**np.ceil(np.log2(nsampl + flen - 1)))
-     sf = np.fft.fft(reference_sources, n=n_fft, axis=1)
-     sef = np.fft.fft(estimated_source, n=n_fft)
+     n_fft = int(2**np.ceil(np.log2(nsampl + flen - 1.)))
+     sf = scipy.fftpack.fft(reference_sources, n=n_fft, axis=1)
+     sef = scipy.fftpack.fft(estimated_source, n=n_fft)
      # inner products between delayed versions of reference_sources
```

## GitHub: Example Workflow

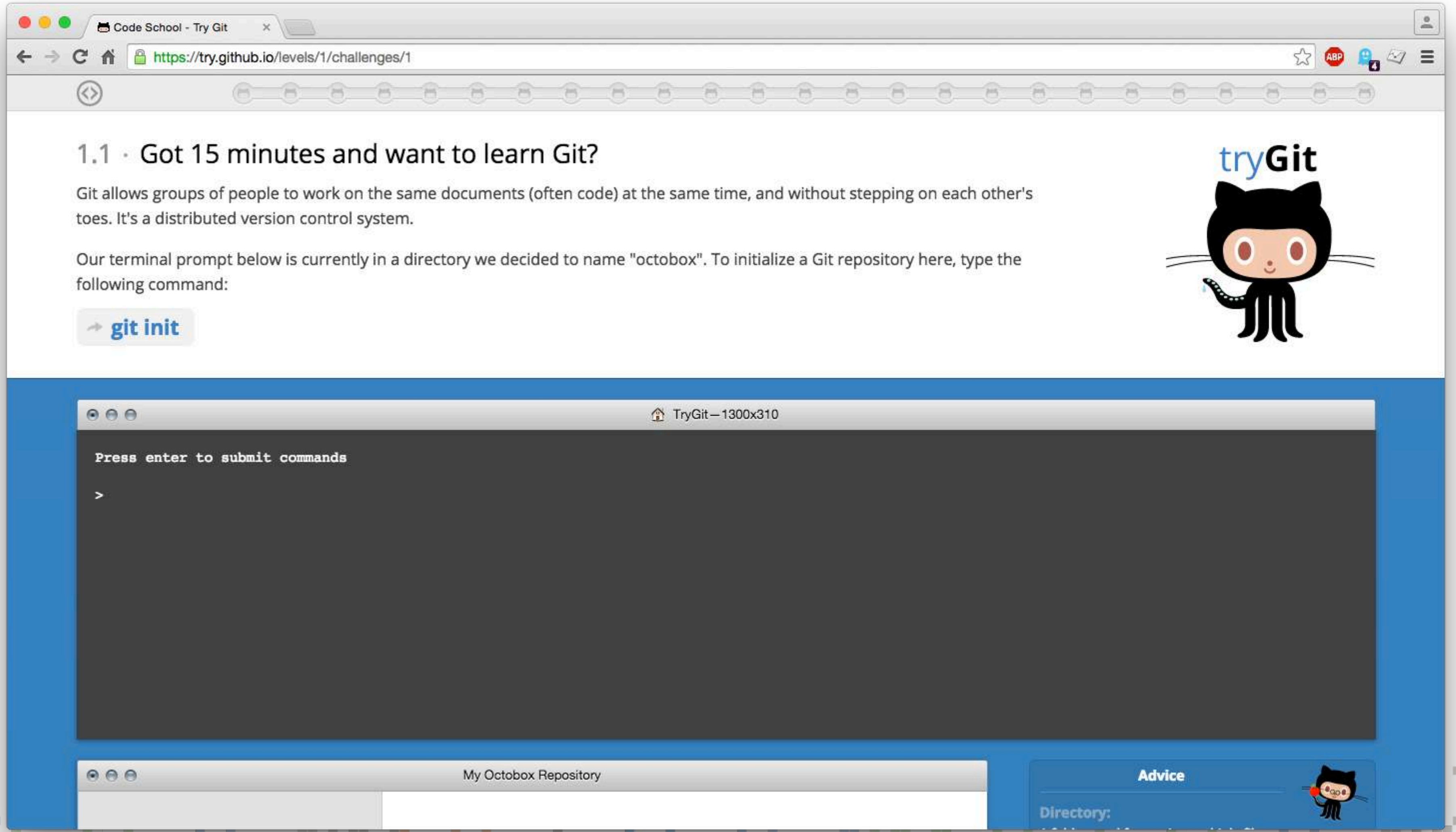
- Pull from GitHub
  - Either `develop` or `master` branch, depends…
- Develop locally
  - First on ipython notebook
  - Then on versioned source
  - Run unit tests
  - Commit
  - Keep editing, pulling changes from collaborators
- When it's ready:
  - Push back to GitHub

## Typical Version Control Questions

- When should I start using version control for my project?
- Which files should I track in the repository?
- How often should I commit?
- How often should I push changes to a shared repo?

## Gitting Started

https://try.github.io/



1.1 · Got 15 minutes and want to learn Git?

Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes. It's a distributed version control system.

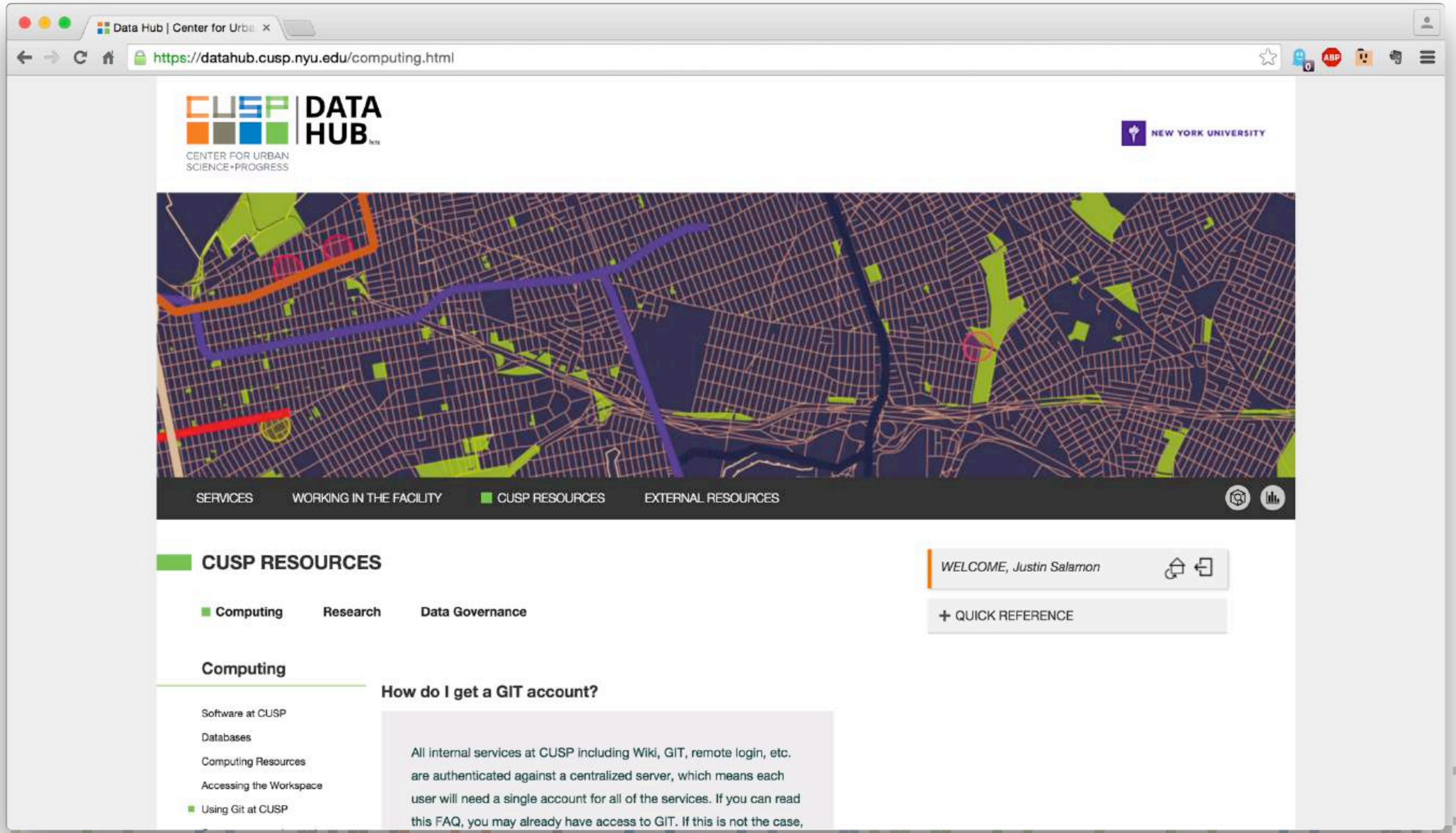
Our terminal prompt below is currently in a directory we decided to name "octobox". To initialize a Git repository here, type the following command:

```
git init
```

```
Press enter to submit commands

>
```

## Gitting Started

https://datahub.cusp.nyu.edu/computing.html



CUSP RESOURCES

Computing

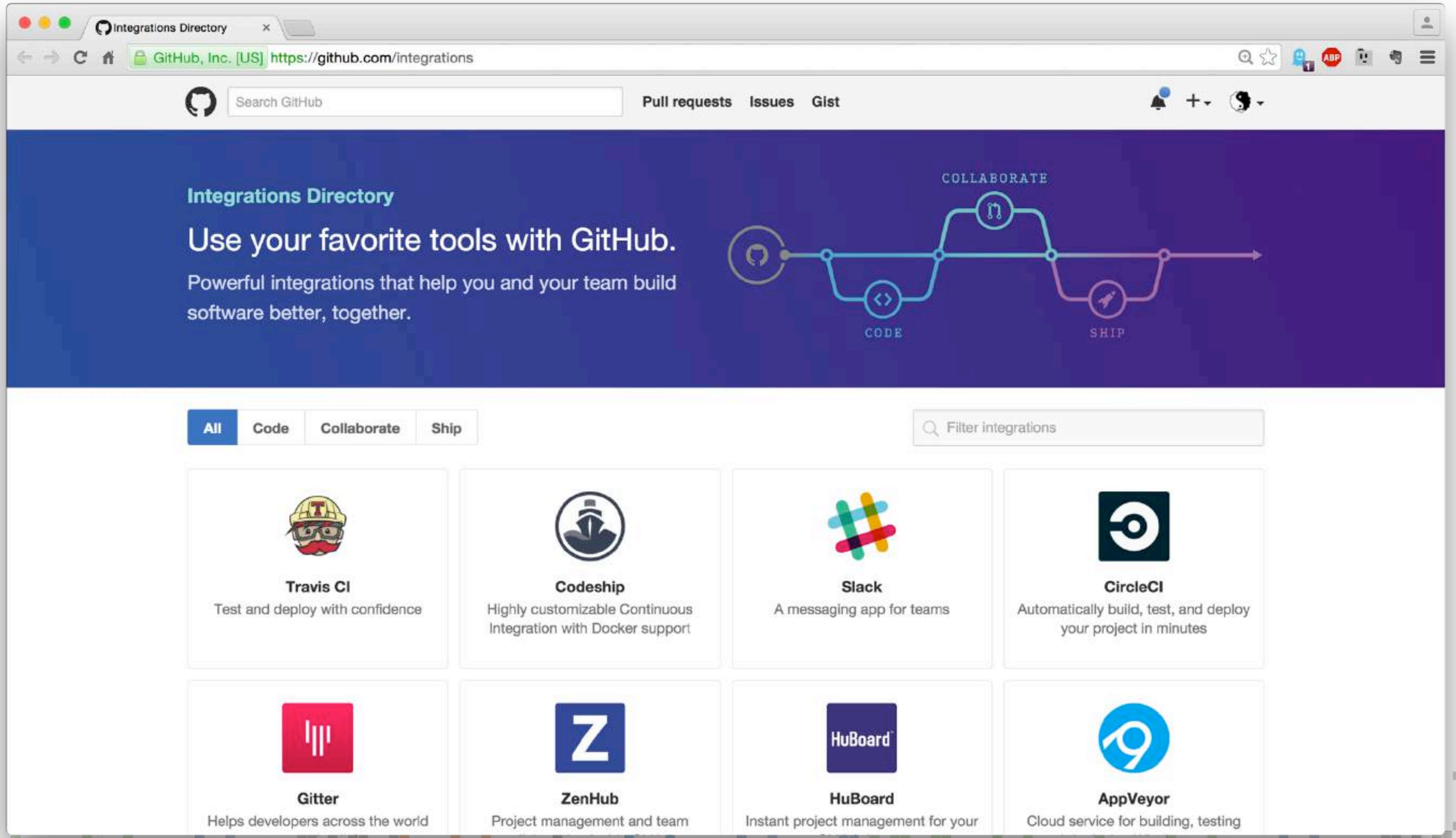
How do I get a GIT account?

All internal services at CUSP including Wiki, GIT, remote login, etc. are authenticated against a centralized server, which means each user will need a single account for all of the services. If you can read this FAQ, you may already have access to GIT. If this is not the case,

## Advanced GitHub

- **Webhooks & services** allow integration of 3rd party "apps" into your GitHub repo:
  - **Travis CI**: continuous integration, runs all unit tests on every pull request, for multiple builds (e.g. python 2.7, 3.4, 3.5)
  - **ReviewNinja**: code review, must get ninja star from someone who's reviewed your code before you can merge PR
  - **Coveralls**: shows percentage of code covered by unit tests, highlights code not covered by any test.
  - Many more…

## Advanced GitHub


Integrations Directory

GitHub, Inc. [US] https://github.com/integrations

Integrations Directory

Use your favorite tools with GitHub.

Powerful integrations that help you and your team build software better, together.

**Travis CI**
Test and deploy with confidence

**Codeship**
Highly customizable Continuous Integration with Docker support

**Slack**
A messaging app for teams

**CircleCI**
Automatically build, test, and deploy your project in minutes

**Gitter**
Helps developers across the world

**ZenHub**
Project management and team

**HuBoard**
Instant project management for your

**AppVeyor**
Cloud service for building, testing

## Publishing Code

- Make sure it has a license!
  - BSD/MIT-style is a good choice for research code
  - GPL for complete applications or code with possible commercial value
  - Ensure license is at least described in a README file
- Make it citable, get a DOI
  - Zenodo.org
  - https://guides.github.com/activities/citable-code/
- Code implementing research?
  - Tell users what they should cite if they use it

## 3 Things To Do Tomorrow

1. Get your current research code into a version control repository & push it to a hosting site (can be private)
2. Pull it onto another computer, get it to build and run
3. Open source? Choose a license!
   – Your code includes someone else's code? Make sure the licenses are compatible!

## Unit Testing

Unit testing is awesome.

We don't have time to cover it.

If you're not familiar with it, look it up.

## Unit Testing: What is it?

- A “unit test” is a bit of code that calls one of your functions, gives it some input, and tells you whether it returned the right result
- Write a set of these, and you have a “test suite”
- A “test framework” can help you write them more quickly; there's at least one for every programming language and environment
- Should be set up so you can run all tests in one go

## Unit Testing: What is it for?

An automated way of ensuring:

- That your code's API works
- That the individual parts of your code work correctly
- That you don't break your code when changing it
- Also useful when developing a tricky algorithm (test-driven development)

## Unit Testing Questions

How do I write tests when I don't know what results to expect?

- Break it down into functions whose behaviour you can predict
- Test individual components, not the whole thing
- Testable code is also more readable code (and so more reviewable code, and…)

Unit testing is about trying to ensure that *the code implements the method*—not that *the method is the right one*

## Unit Testing Questions

What sort of test data and test cases should I write?

- The simplest possible ones!

But I have big data sets and complex results!

- Don’t use real-world data: that’s a different kind of test
- Look for the smallest possible input to test a given behaviour

## Unit Testing Questions

Example…

## How Not To Lose Your Degree

(or: How To Backup Your Data, Share It, and Be Awesome)

## A Show of Hands

## Horror Story 1

*Hi, a friend of mine just overwrote two months of her PhD thesis with an older version. I know recovery of overwritten data is possible, but wonder if I'd need special hardware to do it. Does anyone know something about this ?*

*Thank You.*

5 October 2005 Linux Forums - http://tinyurl.com/8t7uaop

**WORKING COPY IS NOT ENOUGH**

**KEEP BACKUPS**

## Horror Story 2

*A tiny television sits where a big screen used to, and a Christmas tree stands with little underneath it...*

*Even worse than the gifts, the crooks stole a MacBook Pro laptop and a LaCie hard drive.*

*The hard drive had ... her dissertation and nearly seven years of research for her doctoral degree she was set to finish in a few weeks.*

*Osuna had everything backed up on a separate hard drive in a safe, but burglars made off with that too.*

*"All I could think about is that all that time is gone, all that effort, everything is gone," Osuna said.*

**LOCAL COPY IS NOT ENOUGH**

**BACKUP TO THE CLOUD**

*22 December 2010 KRQE - http://tinyurl.com/9a5j56f*

## Horror Story 3

*...her car was broken into and her chrome Mac book pro was stolen. She has a back-up for all but the last six months of research, but the most important part of the research had happened recently.*

NBC4 January 06 2011 - http://tinyurl.com/92jl2lr

**MANUAL BACKUP IS NOT ENOUGH**

**SCHEDULE YOUR BACKUPS**

## Archiving Your Data

- Project (or course, or degree) is over, now what?
- Archive!
  - Allow follow-on research
  - Allow validation of your results

- Beware!
  - Don't use obscure formats
  - Don't use obscure media
    - BBC Domesday Project [1986] used laserdisc!
  - Don't rely on technology being available
  - Keep original source material

## Archiving Your Data

When someone looks at your data, will they understand:

- Why you created it?
- What the data is useful for?
- What column 27 in table 15 actually means?
- What are the UNITS?! (Hertz? Meters? Miles?)
- How the data was created?
- What the source data was on which this data is based?

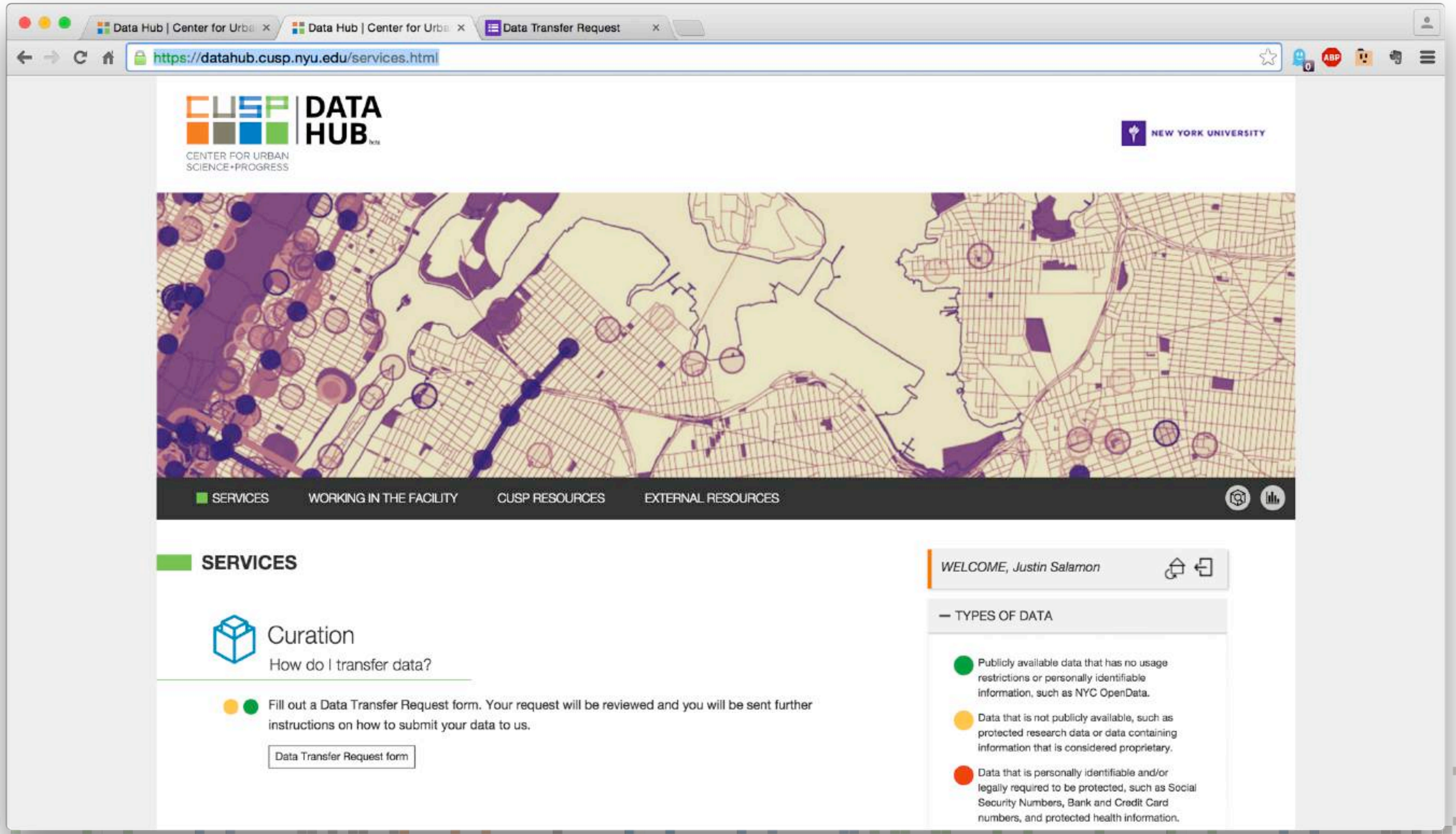
**THERE'S NO DATA WITHOUT METADATA!**

## Archiving Your Data

CUSP Data Facility

## Archiving Your Data

https://datahub.cusp.nyu.edu/services.html



Data Hub | Center for Urba
Data Hub | Center for Urba
Data Transfer Request
https://datahub.cusp.nyu.edu/services.html
CUSP DATA HUB beta
CENTER FOR URBAN SCIENCE+PROGRESS
NEW YORK UNIVERSITY
SERVICES
WORKING IN THE FACILITY
CUSP RESOURCES
EXTERNAL RESOURCES
WELCOME, Justin Salamon
Data Transfer Request form

SERVICES

Curation

How do I transfer data?

● ● Fill out a Data Transfer Request form. Your request will be reviewed and you will be sent further instructions on how to submit your data to us.

— TYPES OF DATA

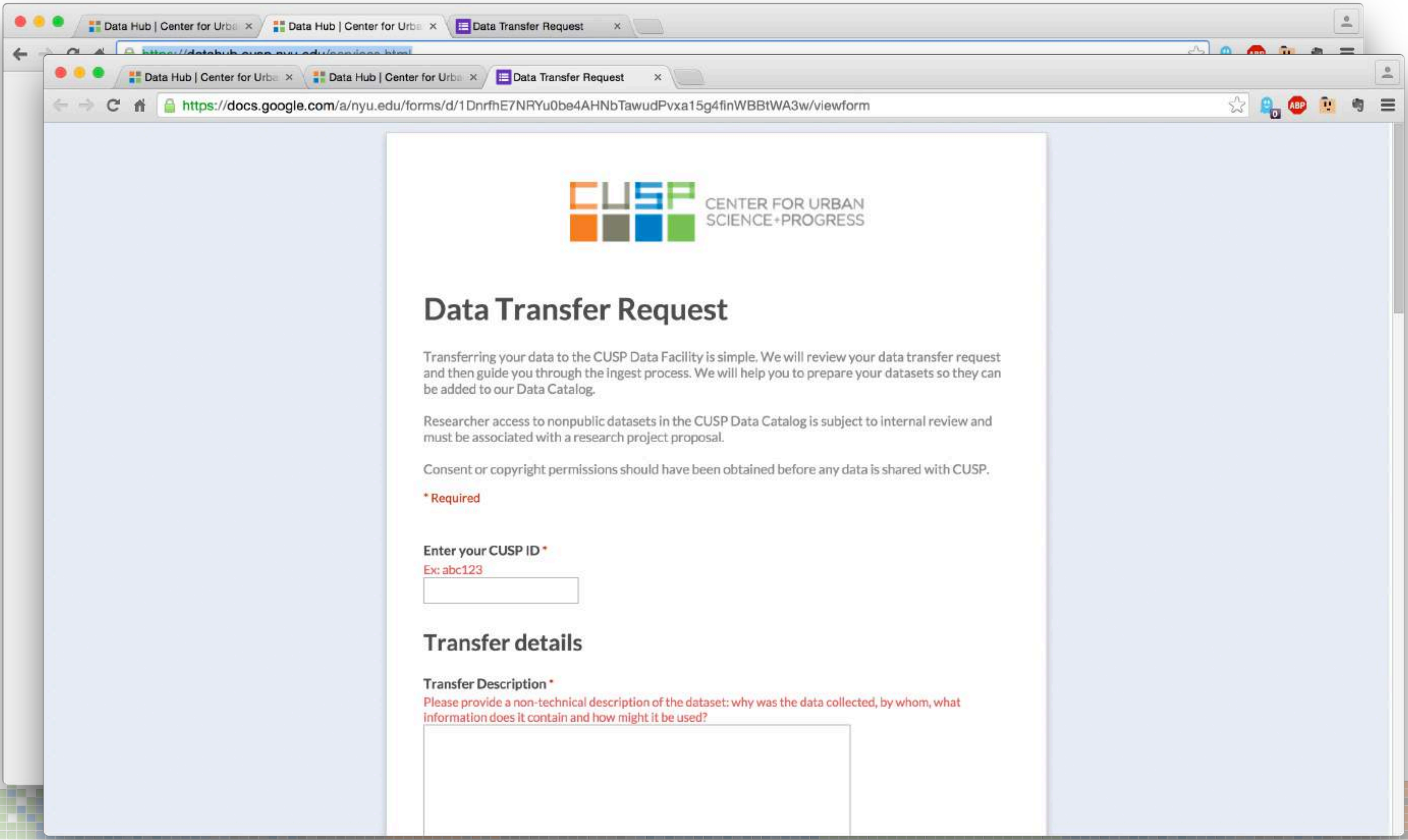
● Publicly available data that has no usage restrictions or personally identifiable information, such as NYC OpenData.

● Data that is not publicly available, such as protected research data or data containing information that is considered proprietary.

● Data that is personally identifiable and/or legally required to be protected, such as Social Security Numbers, Bank and Credit Card numbers, and protected health information.

## Archiving Your Data

https://datahub.cusp.nyu.edu/services.html



Data Hub | Center for Urba
Data Hub | Center for Urba
Data Transfer Request
https://docs.google.com/a/nyu.edu/forms/d/1DnrfhE7NRYu0be4AHNbTawudPvxa15g4finWBBtWA3w/viewform
CUSP CENTER FOR URBAN SCIENCE+PROGRESS

Data Transfer Request

Transferring your data to the CUSP Data Facility is simple. We will review your data transfer request and then guide you through the ingest process. We will help you to prepare your datasets so they can be added to our Data Catalog.

Researcher access to nonpublic datasets in the CUSP Data Catalog is subject to internal review and must be associated with a research project proposal.

Consent or copyright permissions should have been obtained before any data is shared with CUSP.

* Required

Enter your CUSP ID *

Ex: abc123

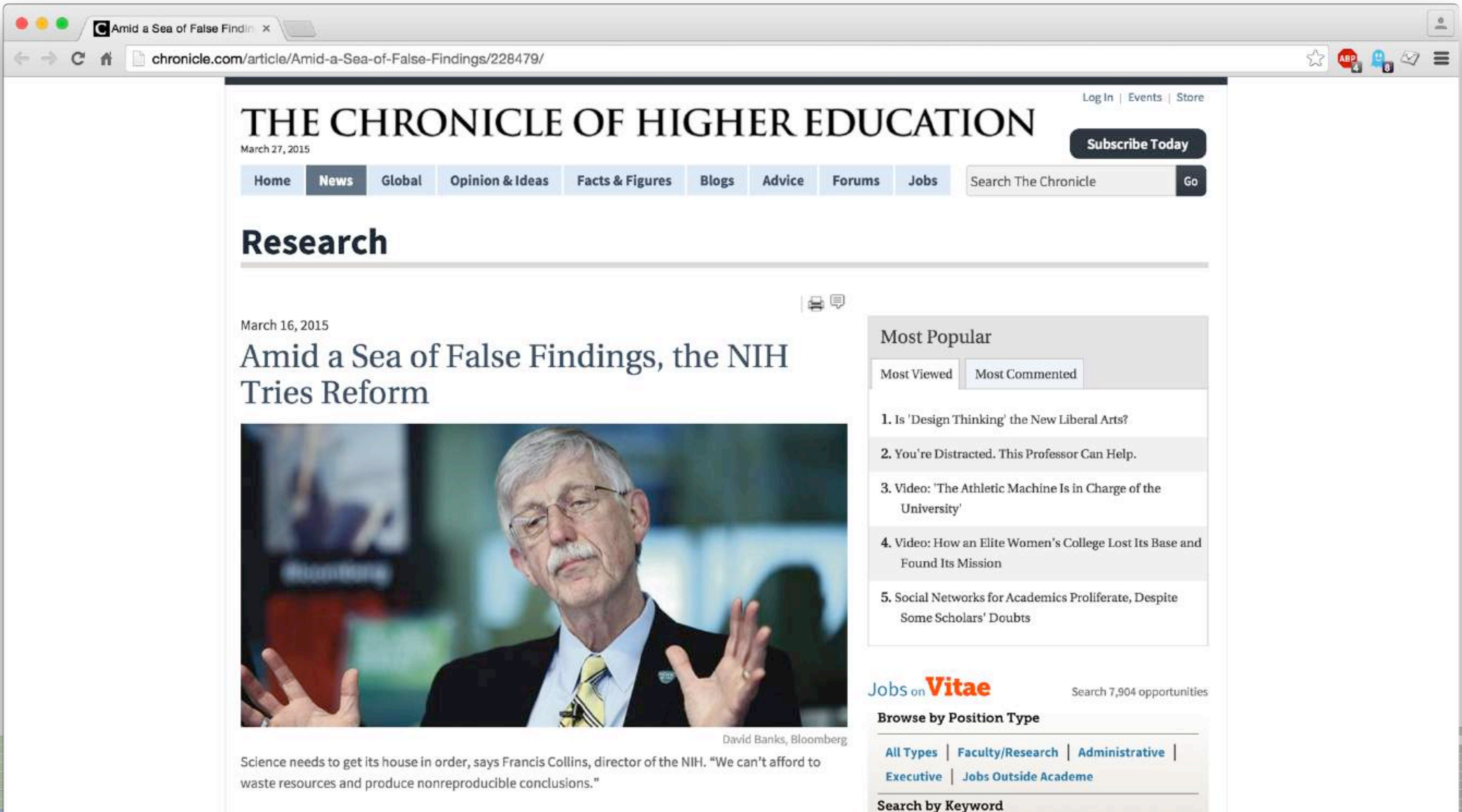
Transfer details

Transfer Description *

Please provide a non-technical description of the dataset: why was the data collected, by whom, what information does it contain and how might it be used?

## How Not To Lose Your Degree: Publishing Data

"Many researchers believe that if scientists set out to reproduce preclinical work published over the past decade, a majority would fail. This, in short, is the reproducibility crisis."



THE CHRONICLE OF HIGHER EDUCATION

March 27, 2015

Research

March 16, 2015

Amid a Sea of False Findings, the NIH Tries Reform

David Banks, Bloomberg

Science needs to get its house in order, says Francis Collins, director of the NIH. "We can't afford to waste resources and produce nonreproducible conclusions."

Most Popular

Most Viewed | Most Commented

1. Is 'Design Thinking' the New Liberal Arts?
2. You're Distracted. This Professor Can Help.
3. Video: 'The Athletic Machine Is in Charge of the University'
4. Video: How an Elite Women's College Lost Its Base and Found Its Mission
5. Social Networks for Academics Proliferate, Despite Some Scholars' Doubts

Jobs on Vitae

Search 7,904 opportunities

Browse by Position Type

All Types | Faculty/Research | Administrative | Executive | Jobs Outside Academe

Search by Keyword

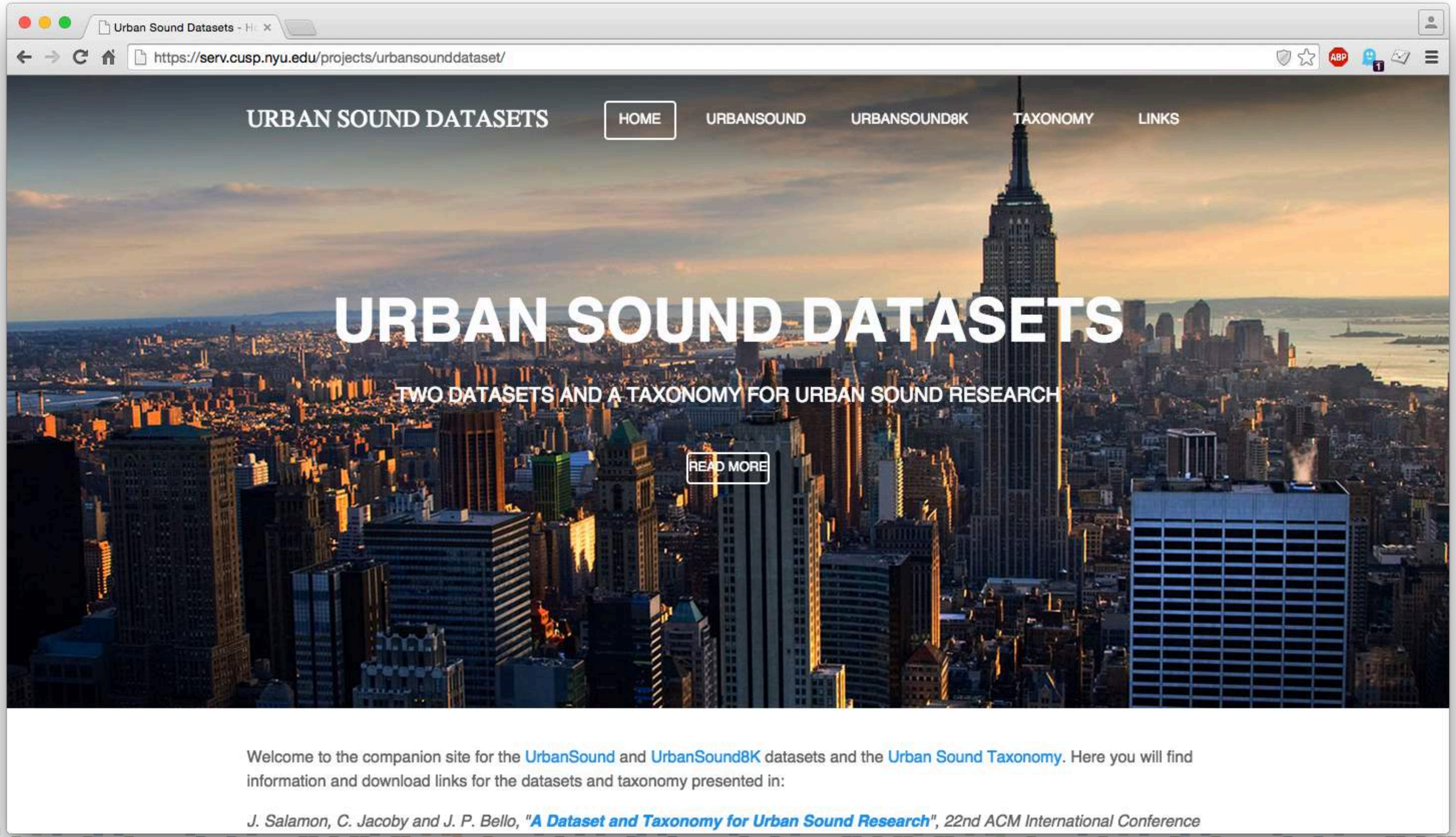
## Publishing Your Data

Why publish my data?

- Help others to:
  - Validate your research
  - Validate their implementation of your algorithm
  - Advance the state of the art
  - Combine with other datasets

- So what's in it for me?
  - Citations!
  - Impact! (job)

## Publishing Your Data



Welcome to the companion site for the UrbanSound and UrbanSound8K datasets and the Urban Sound Taxonomy. Here you will find information and download links for the datasets and taxonomy presented in:

*J. Salamon, C. Jacoby and J. P. Bello, "**A Dataset and Taxonomy for Urban Sound Research**", 22nd ACM International Conference*

## Publishing Your Data

- Make sure you are allowed to publish!
  - Who owns the data?
    - NYU / Industry partner / Funding body / Creative Commons
  - Adhere to CUSP policies
    - Privacy: are you allowed to publish the data?
    - Publication: are you expected to publish the data?
    - Repositories: where should you publish the data?
    - Licenses: who should be allowed to access the data?
- Make sure it has a license!
  - Example: Creative Commons
    - CC0? CC Attribution Non-Commercial? Something else?

# How Not To Lose Your Future Job

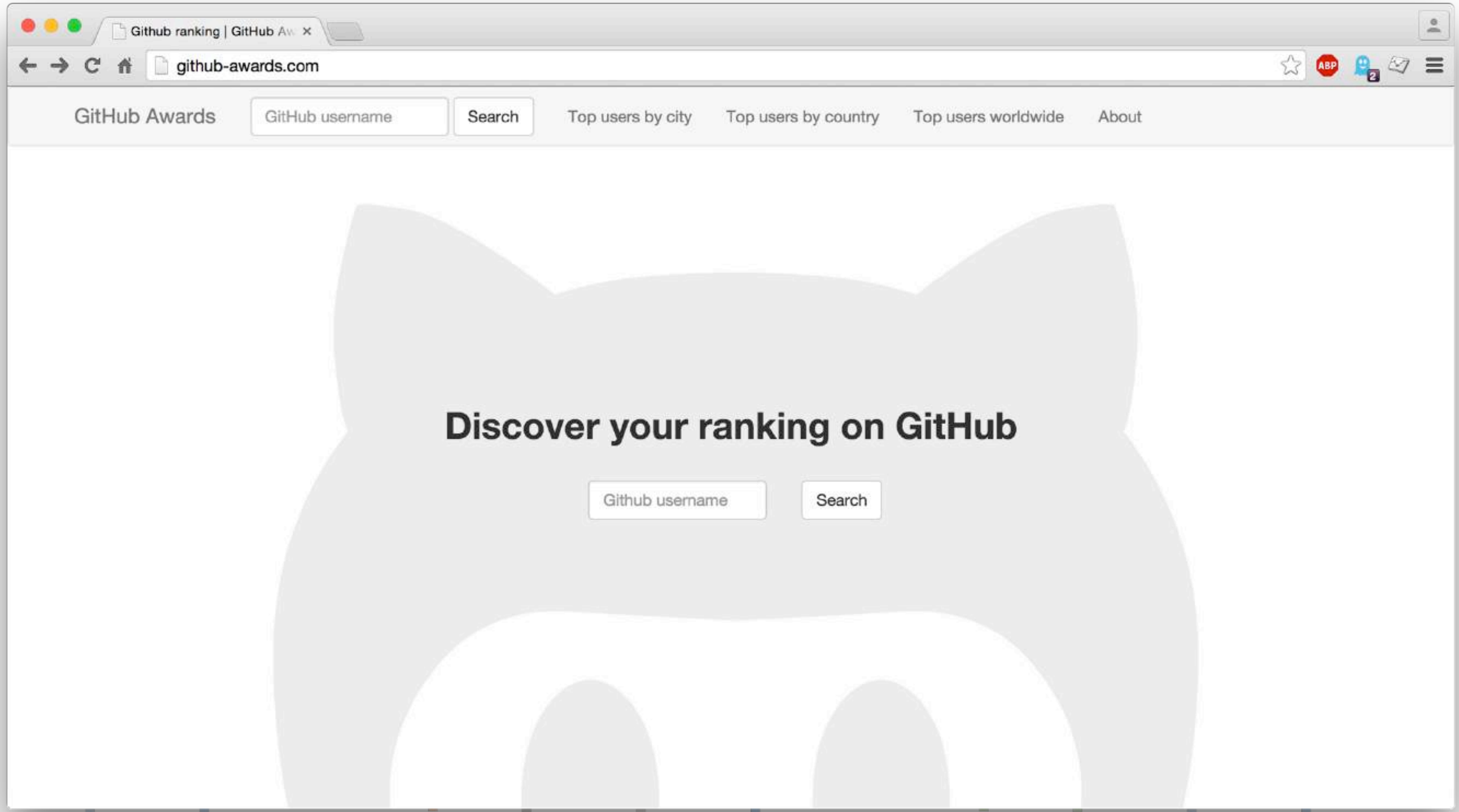
(or: How To Maximize The Impact Of  Your Work and Be Awesome)

## Get Found

## Git Found?

- Employers/recruiters increasingly searching/asking for GitHub profiles
- But wait… there's more…

## Git Found?



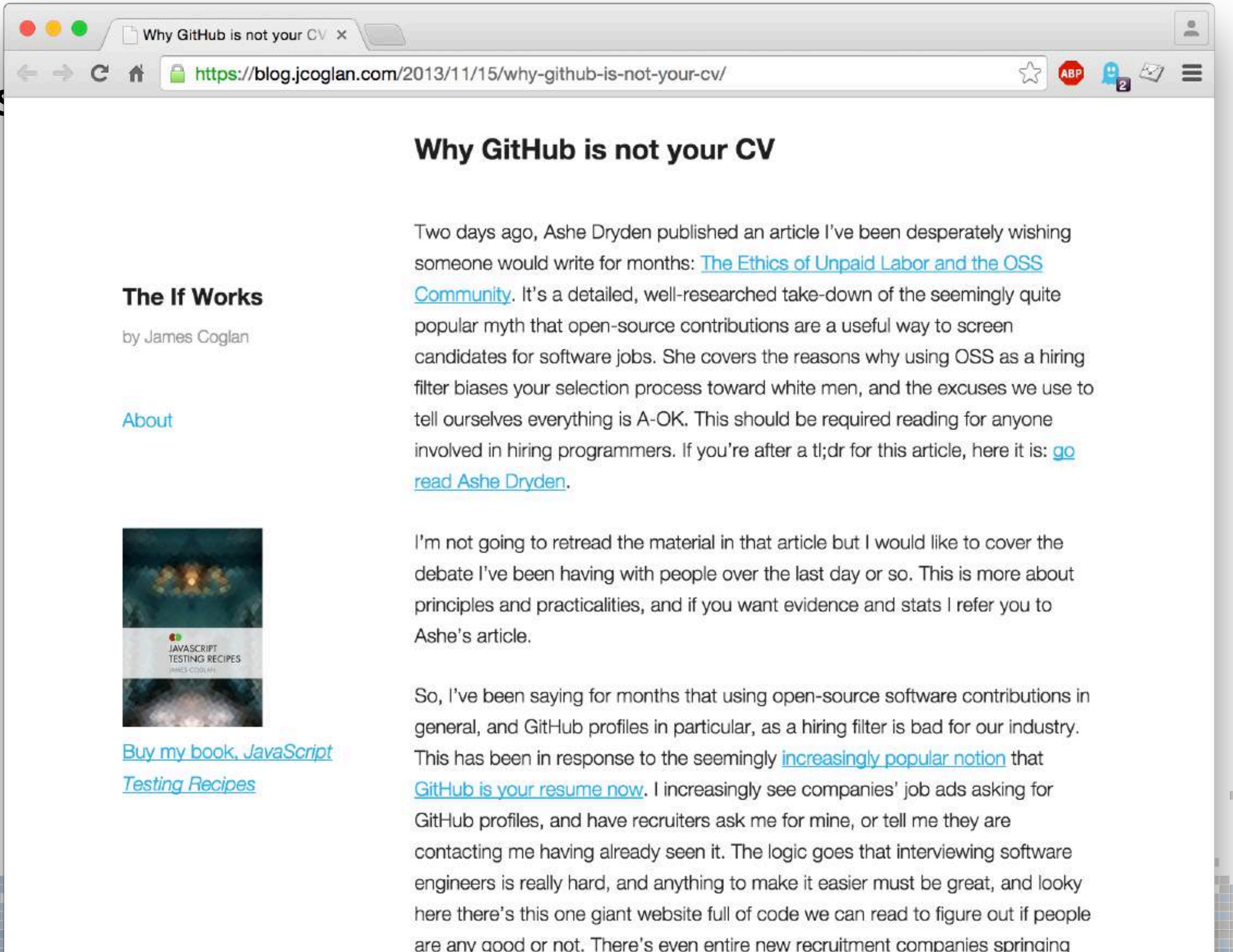
Discover your ranking on GitHub

## Git Found?

So is GitHub my new CV?

## Git Found?

So is



Why GitHub is not your CV

https://blog.jcoglan.com/2013/11/15/why-github-is-not-your-cv/

**The If Works**

by James Coglan

About

JAVASCRIPT TESTING RECIPES

JAMES COGLAN

Buy my book, *JavaScript Testing Recipes*

**Why GitHub is not your CV**

Two days ago, Ashe Dryden published an article I've been desperately wishing someone would write for months: The Ethics of Unpaid Labor and the OSS Community. It's a detailed, well-researched take-down of the seemingly quite popular myth that open-source contributions are a useful way to screen candidates for software jobs. She covers the reasons why using OSS as a hiring filter biases your selection process toward white men, and the excuses we use to tell ourselves everything is A-OK. This should be required reading for anyone involved in hiring programmers. If you're after a tl;dr for this article, here it is: go read Ashe Dryden.

I'm not going to retread the material in that article but I would like to cover the debate I've been having with people over the last day or so. This is more about principles and practicalities, and if you want evidence and stats I refer you to Ashe's article.

So, I've been saying for months that using open-source software contributions in general, and GitHub profiles in particular, as a hiring filter is bad for our industry. This has been in response to the seemingly increasingly popular notion that GitHub is your resume now. I increasingly see companies' job ads asking for GitHub profiles, and have recruiters ask me for mine, or tell me they are contacting me having already seen it. The logic goes that interviewing software engineers is really hard, and anything to make it easier must be great, and looky here there's this one giant website full of code we can read to figure out if people are any good or not. There's even entire new recruitment companies springing

## Git Found?

- Even a stellar GitHub account is not a replacement for:
  - Your resume
  - Interview skills
  - Presentation skills

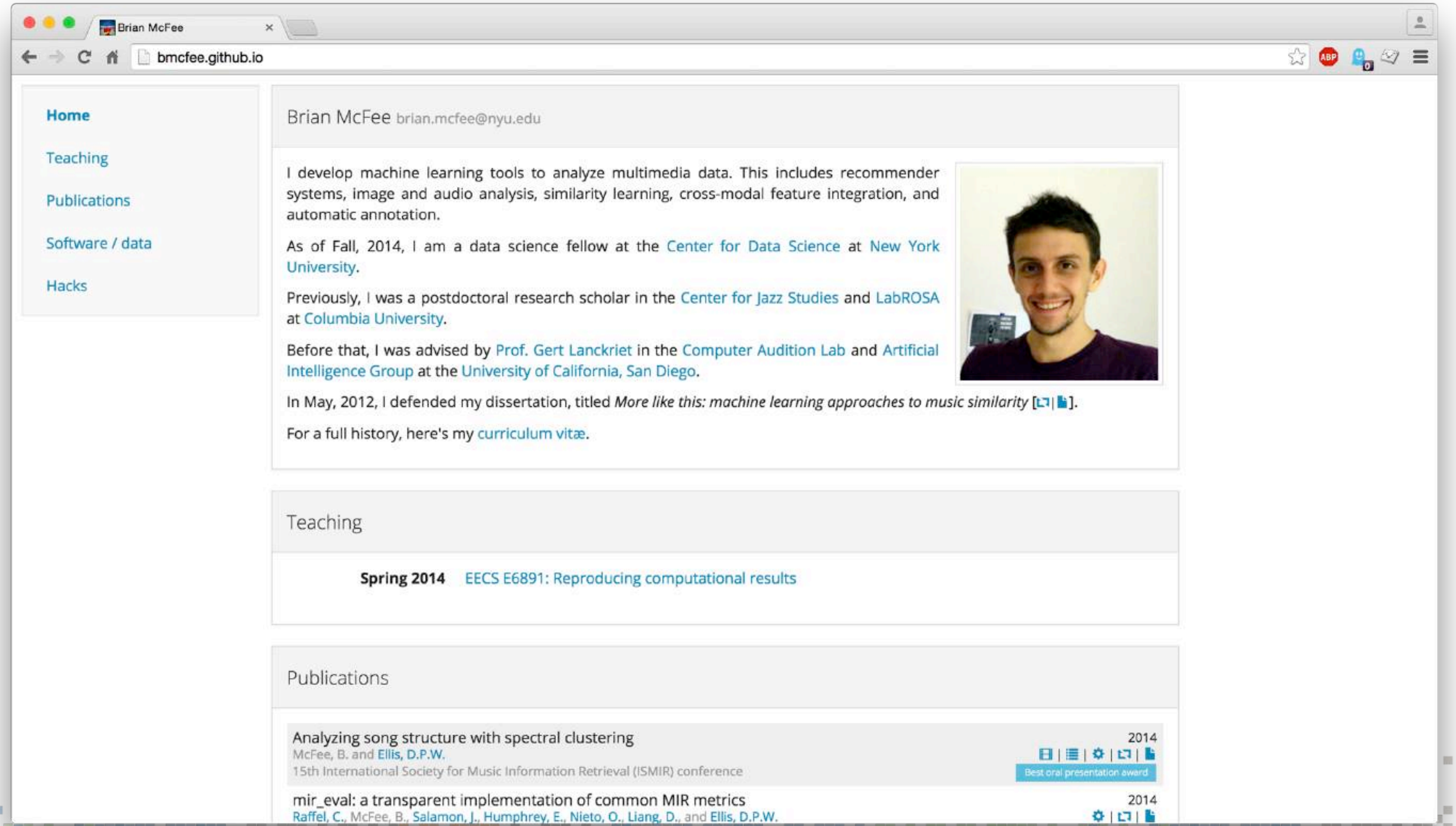
- But… it can help you get the interview in the first place!
- Using Git (not specifically GitHub) is a basic skill every programmer should have nowadays

## Take Control of Your Online Presence

- LinkedIn, GitHub, etc.
  - Limited to displaying a specific type of information
- A personal website is where you can really showcase your best!

How Not To Lose Your Future Job: Online presence

# Yes



Home

Teaching

Publications

Software / data

Hacks

Brian McFee brian.mcfee@nyu.edu

I develop machine learning tools to analyze multimedia data. This includes recommender systems, image and audio analysis, similarity learning, cross-modal feature integration, and automatic annotation.

As of Fall, 2014, I am a data science fellow at the Center for Data Science at New York University.

Previously, I was a postdoctoral research scholar in the Center for Jazz Studies and LabROSA at Columbia University.

Before that, I was advised by Prof. Gert Lanckriet in the Computer Audition Lab and Artificial Intelligence Group at the University of California, San Diego.

In May, 2012, I defended my dissertation, titled *More like this: machine learning approaches to music similarity* [ | ].

For a full history, here's my curriculum vitæ.

Teaching

**Spring 2014** EECS E6891: Reproducing computational results

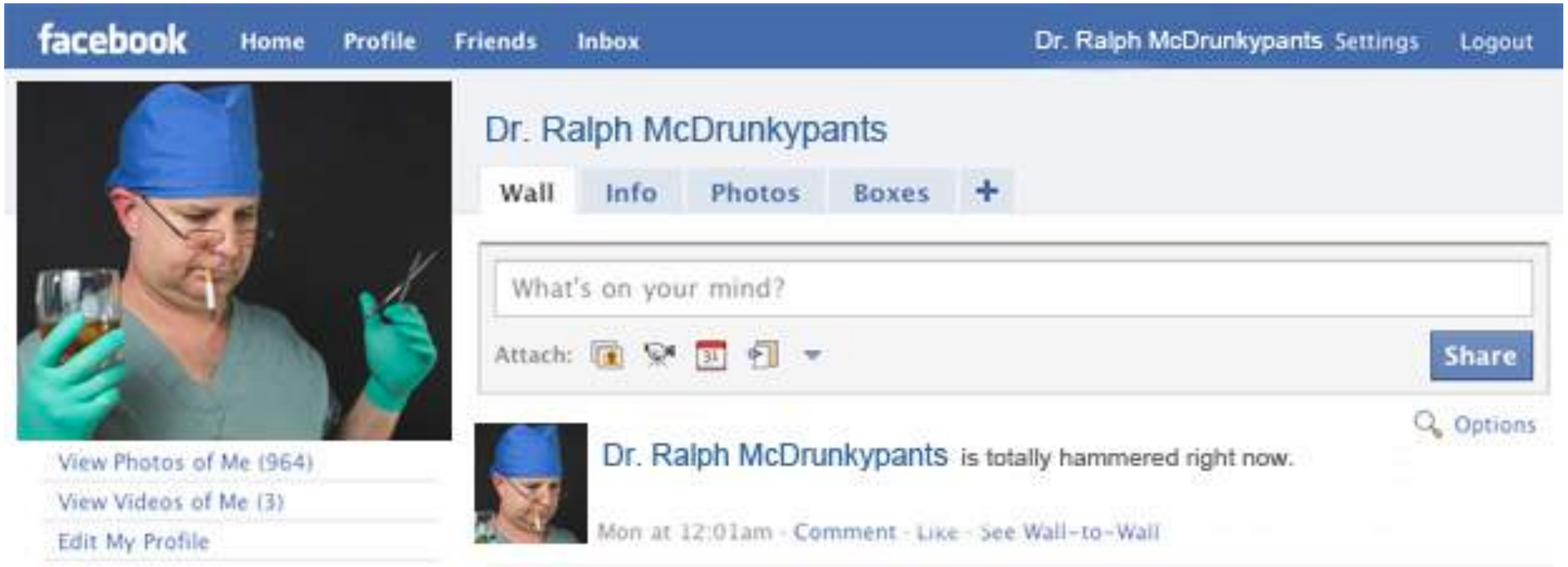
Publications

Analyzing song structure with spectral clustering 2014
McFee, B. and Ellis, D.P.W.
15th International Society for Music Information Retrieval (ISMIR) conference
Best oral presentation award

mir_eval: a transparent implementation of common MIR metrics 2014
Raffel, C., McFee, B., Salamon, J., Humphrey, E., Nieto, O., Liang, D., and Ellis, D.P.W.

How Not To Lose Your Future Job: Online presence

## No



Dr. Ralph McDrunkypants

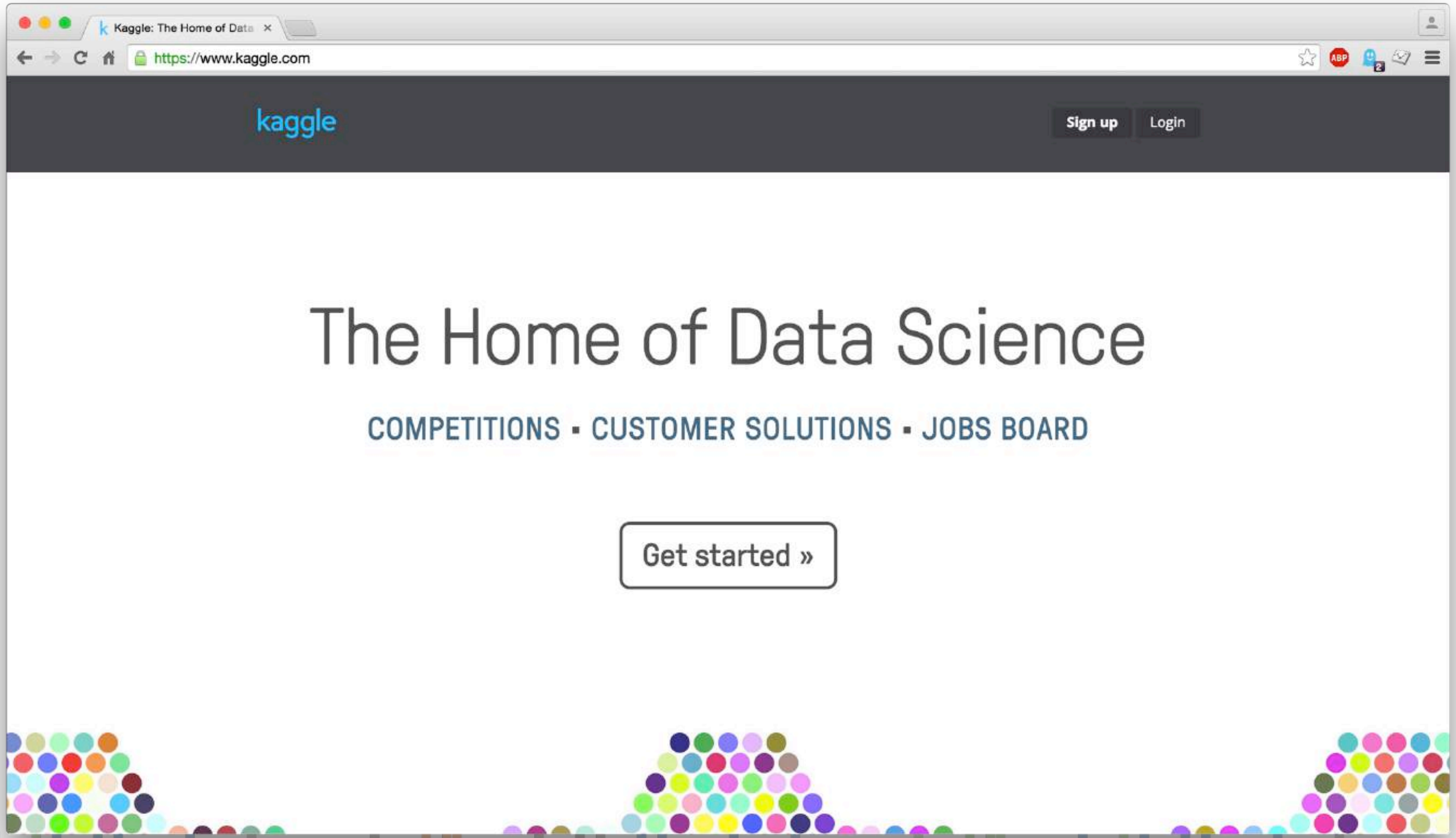
Dr. Ralph McDrunkypants is totally hammered right now.

Mon at 12:01am · Comment · Like · See Wall-to-Wall

## Personal Website

- Showcase your best
- If an employer searches for you (online), have control over the first thing they see
- An interesting blog-post (or ipython notebook!) can be a great source of traffic to your site!

## Kaggle



The Home of Data Science

COMPETITIONS · CUSTOMER SOLUTIONS · JOBS BOARD

## Kaggle

- Excellent experience…
- But will it land me a job?
  - If you win…

## Credits

Parts of this presentation (version control, GitHub, unit testing, data publishing/archiving and some horror stories) taken from:

- Chris Cannam et. al (Queen Mary, University of London): ISMIR 2012 tutorial on Reusable software and reproducibility in music informatics research
  - http://soundsoftware.ac.uk/videos
- Dan Ellis & Brian McFee (Columbia): Version Control & Github:
  - http://www.ee.columbia.edu/~dpwe/e6891/outline.html

## RECAP

- Code
  - Use version control (probably Git)
  - Write unit tests
  - Use online hosting (e.g. GitHub / BitBucket)
- Data
  - Back it up!
  - Publish it / archive it
- Online presence
  - Great way to get attention from employers
    - (but not a replacement for the classic skill-set)
  - Be in control of your online presence!

Q&A Time