Software development best-practises for reproducible research

Alastair Porter

20 September 2018

MIP-Frontiers Meeting

I'm Alastair

- I'm a researcher in the MTG
- I do research, but the majority of my time is spent writing software
- I have experience working in large organisations, and in community driven projects with hundreds of thousands of users
- This talk is based on my experience writing software in a research context, and questions that I get asked by students

The ideal research pipeline

Researcher A ("Producer")

- Read background papers
- Do own research
- Publish paper X

Researcher B ("Consumer-Producer")

- Read paper X
- Understand/reproduce results in paper X
- Do more research building on X
- Publish paper Y that cites X / produce product that uses X

Plumbley et al. 2012. Research reproducibility workshop

The real research pipeline

Researcher A (“Producer”)

- Read background papers
- Do own research (including lots of coding)
- Publish paper X (no source code or data. May describe the algorithm incorrectly)

Researcher B (“Consumer-Producer”)

- Read paper X
- Can’t reproduce or use results in paper X
- Tear out hair
- Give up / do something else

Plumbley et al. 2012. Research reproducibility workshop

Research is software development

- Papers need code

- How many of you are programming as part of your research?
- How many of you did an undergraduate degree in CS or Software Development?
- How many hadn't programmed before you started here?

Overview of this talk

1. Code layout and organisation
2. Testing
3. Documentation
4. Source control
5. Data
6. Licensing

- Not a tutorial talk.
  - You will have to take these ideas and investigate them further
- Time for questions at the end, but please stop me if you want to discuss something

A small disclaimer

I'm not very good at this

- 2014 & 2015: I gave a talk similar to this one, about best practises in making research reproducible
- 2015: Did an experiment submitted a paper to a conference
- Paper was rejected
- 2016: I wanted to re-visit the paper, change some processes, and re-submit

- Should be easy, right? Just read the documentation and run the right script

No

I didn’t even know what file to run to reproduce the results, let alone where to start making changes

Listening to a talk will not make you better

- That is to say, this is not magic
- You can't just decide to write good software and suddenly it will be good
- Like everything else, pick a small change to make, practise it until it's second nature. Repeat

There is a happy ending

AudioCommons Jamendo tools

This project contains tools used to get metadata and recordings from Jamendo for use in the AudioCommons project.

We are currently working with the Jamendo Licensing catalog. This catalog has about 200,000 available tracks, of which we are using a subset of about 80,000.

Setup

This code uses python 3. Make sure to use a python3 virtual environment (`virtualenv -p python3.4 ve`) or use `python3` / `pip3` .

Install dependencies with

```
$ pip -r requirements
```

To configure, copy `config.py.dist` to `config.py` and fill in your jamendo api key.

Usage

To get the contents of the licensing catalog

```
$ python getlicensing.py <outputname.json>
```

To update the catalog with entries that have been published since a playlist was downloaded

```
$ python getlicensing.py -u <existingplaylist.json> <outputname.json>
```

To get the contents of a playlist

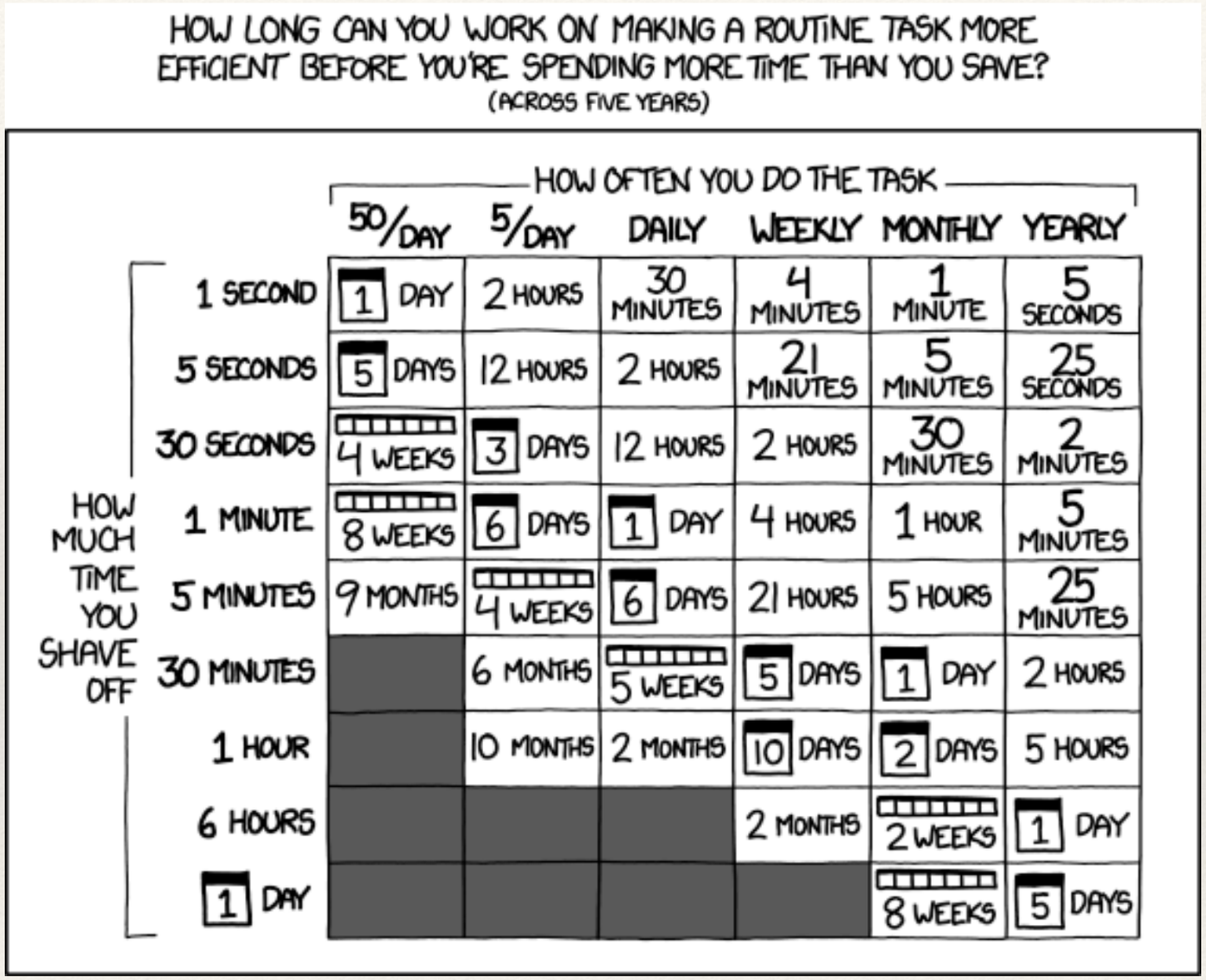
I'm here to transfer knowledge

- Observations and things that I've learned from 10 years of programming and software development
- I will list some main points which have helped me through the years, why they're a good idea, and why they've worked for me
- Almost all of these points will require more up-front work.

"But I'm a very busy person"

- Your first thought will probably be "But this is going to take much longer. I don't have time for this, I have a paper due really soon!"
- Additional effort up front will save you time in the future

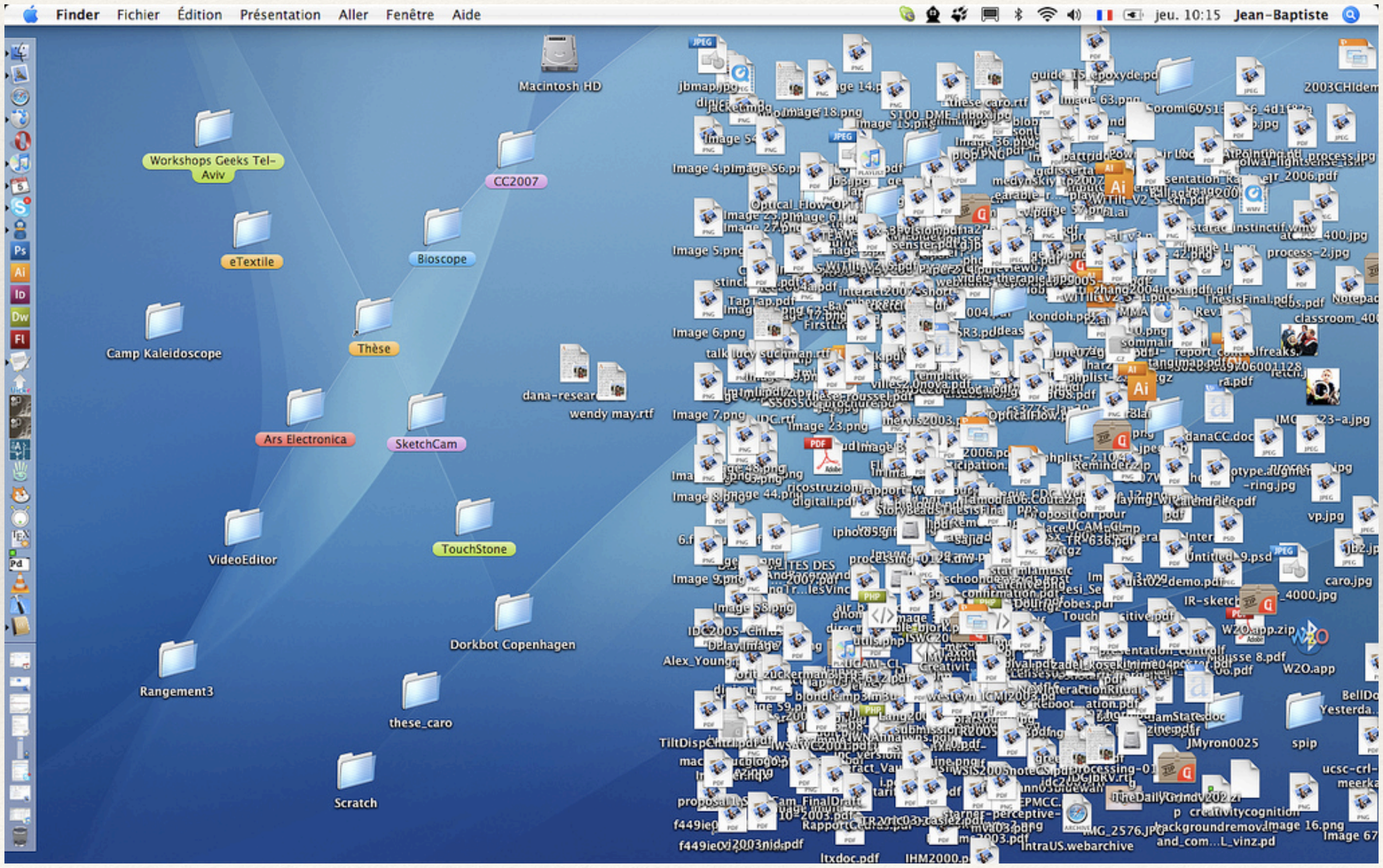
I'm a very busy person



https://xkcd.com/1205/

Starting a new project

- You have an idea for a project. Great, let's start

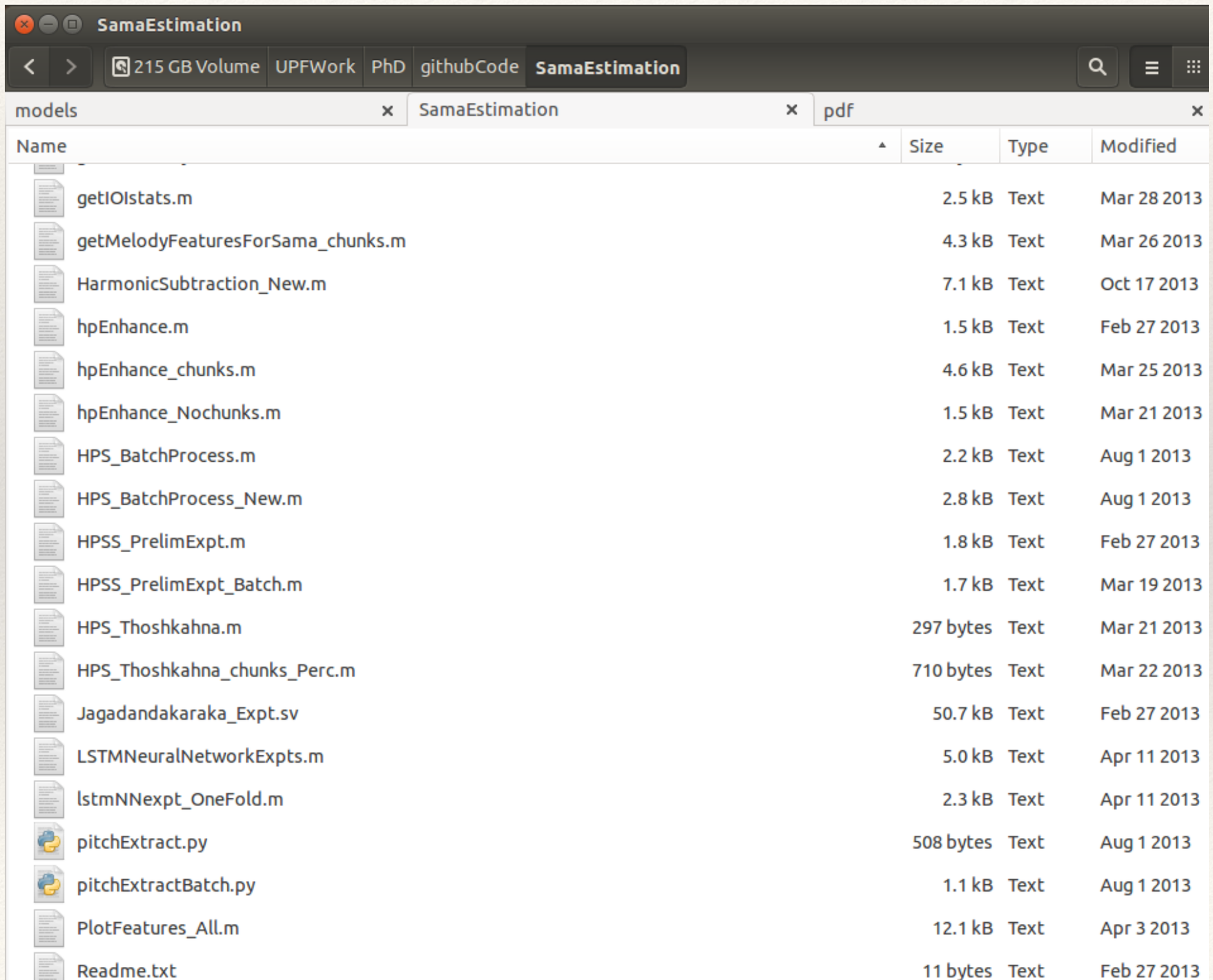


https://www.flickr.com/photos/jeanbaptisteparis/724619122/

Advice from the Internet

https://twitter.com/micahgallen/status/1001362580710088704
https://twitter.com/ProfZare/status/1001424807429275649

Code layout and organisation



Code layout and organisation

- what is hpEnhance_chunks? _nochunks?
- What is normal hpEnhance?
- What is BatchProcess_New? What does it do new which is different?
- Don’t copy an entire file and rename it to try something new
  - You know what these files do now, but will you in 6 months?
  - What if you find an error in your evaluation component? Will you remember to change the original code and test that too?

Breaking up code into units

- Don't put all your code into one method in one file
- Why not? - It makes it more difficult to reason about parts
- makes testing harder (more on this later)
- makes reuse harder

Breaking up code into units

- Split code into smaller blocks
  - Load
  - Preprocess
  - Main algorithm
  - Calculate statistics
- Change the algorithm? Make a second function instead of copying all additional code too
- Have a runtime flag (or setting variable) which lets you choose which algorithm to use

Breaking up code into units

- How does this save time?

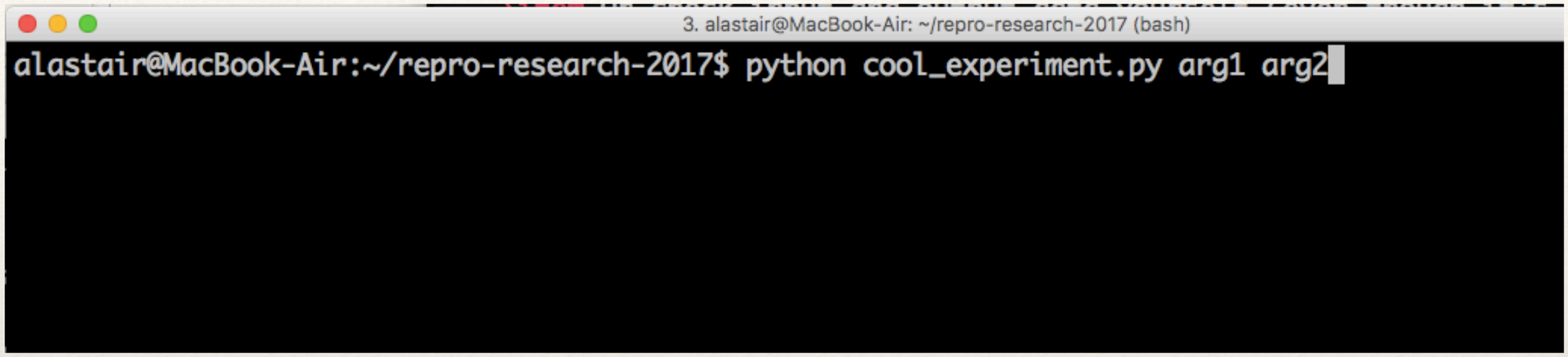
- More up-front work to split code into modules (although, not much more if you split it from the beginning)
- You only have one version of common code
- Save time having to look through each of your versions to find the most up-to-date one
- Common interface to all of your versions of code—Choose between them quickly for evaluation

Hard-coded paths

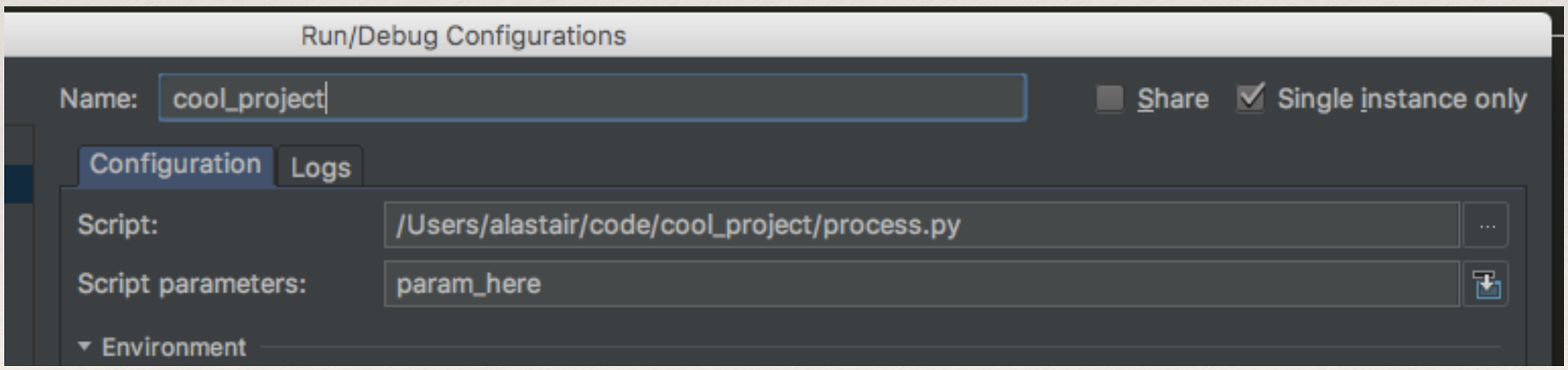
```
20  def query():
21      g = rdflib.Graph()
22      mo = g.parse("C:\Users\Sergio\Dropbox\QMUL\Data\searchOntology.owl")
23      for subj, pred, obj in mo:
24              moSub.append(subj)
25              moPred.append(pred)
26              moObj.append(obj)
27              print "Subject: " + subj
28              print "Predicate: " + pred
29              print "Object: " + obj
```

- Not portable off your computer (very bad if someone else wants to run it)
- Use arguments instead

Using arguments



- Pass arguments when you run your tool



- Using an IDE? You can still add arguments

Using arguments

```
if __name__ == "__main__":
    myfunction(sys.argv[1], sys.argv[2], sys.argv[3])
```

- In python, don't read sys.argv

```
if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Process a datafile into parts")
    parser.add_argument("-d", action="store_true", help="Print debugging info")
    parser.add_argument("input", help="Input json file")
    parser.add_argument("result_dir", help="Result directory to write items")
    args = parser.parse_args()
    myfunction(args.input, args.result_dir, args.verbose)
```

- Use argparse (but it's so complex…)

Argparse magic

- But we want our work to be understandable

1. alastair@MacBook-Air: ~/repro-research-2017 (bash)

```
alastair@MacBook-Air:~/repro-research-2017$ python convert_metadata.py --help
usage: convert_metadata.py [-h] [-d] input result_dir

Process a datafile into parts

positional arguments:
  input       Input json file
  result_dir  Result directory to write items

optional arguments:
  -h, --help  show this help message and exit
  -d          Print debugging info
alastair@MacBook-Air:~/repro-research-2017$
```

- You get documentation for free

Other options for Arguments

- Fire (https://github.com/google/python-fire)
- Magically creates help text and argument handling from a python object or model

Here's an example of calling Fire on a class.

```
import fire

class Calculator(object):
  """A simple calculator class."""

  def double(self, number):
    return 2 * number

if __name__ == '__main__':
  fire.Fire(Calculator)
```

Then, from the command line, you can run:

```
python calculator.py double 10  # 20
python calculator.py double --number=15  # 30
```

Arguments

- How does this save time?
- Less work when you want to release it publicly
- If you need to run the software on another machine, less work to configure it (pass all configuration options as arguments)
- Faster to deploy on the HPC

Testing and Automation

- We have broken our code into small blocks
- How do we make sure the blocks work as they should?
- Tests!
- Are tests worth the effort?
- That depends on who you ask and what your definition of testing is

“Writing tests takes too long”

- Yes, it does.
- In software development, it’s not unusual that writing tests takes as long as writing the software. This means that your development time could **double**
- So, why is it good?
- We are saving time when making changes or checking code

Typical development workflow

1. Make a change
2. Restart your python terminal
3. import your module
4. open a file
5. forget to import json module
6. import json. open file again
7. get a field from json file
8. call your method with data as an argument
9. print out result
10. see that the formatting of your output isn't right
11. go to step 1

Easier development

1. Make a change
2. run python mycode/test_parser.py
3. Fix any errors
4. Repeat

Testing as automation

- Rule of thumb: If you're doing something manually more than 2 or 3 times, perhaps you should write a test or automate it

- Yes, you could press the up button in ipython. It's still slower than writing a test
- If you come back to your code in 3 weeks time can you remember how you tested it?

Testing as documentation

- A test is a kind of documentation for how to use a system (or a method)
- Sometimes when you write that documentation you realise the interface to your code is going to be unnecessarily clumsy
- Change your code to make it easier to automate and to document

Testing

- How does this save time?

- Test only a small part of your algorithm instead of waiting for the entire load/process stage to complete
- Influences the design of small connectable blocks of code, potentially saving refactoring time in the future

Testing example

- There's a bug in this code

In [ ]:

```
def filelist_for_makam_and_form(fileList, makam, form):
    '''Selects/returns a file list (a subset of the whole list: 'fileList')
    given the filters: makam(str) and form(str)'''
    selectedFiles = []
    for filepath in fileList:
        filename = os.path.basename(filepath)
        if filename.split('--')[0] == makam and filename.split('--')[1] == filename:
            selectedFiles.append(filepath)
    return selectedFiles

# Let's pick two makams only differing in seyir and plot melodic curves
# for first sections in the sazsemaisi form
makams = ['rast', 'mahur']
form = 'sazsemaisi'

# Example
txtFilesList = ['muhayyer--bozlak--serbest--havayi_da--asaf_guven.txt',
                'ussak--sarki--nimsofyan--gordum_seni--sadettin_kaynak.txt',
                'mahur--sazsemaisi--aksaksemai----nikolaki.txt',
                'sipihr--pesrev--muhammes----dilhayat_kalfa.txt',
                'mahur--sazsemaisi--aksaksemai--bahar_1--goksel_baktagir.txt',
                'buselik--sarki--duyek--sevgim_bitti_mi--arif_sami_toker.txt']

for index, makam in enumerate(makams):
    selectedFileList = filelist_for_makam_and_form(txtFilesList, makam, form)
    print('Files in makam {} and form {}'.format(makam, form))
    print(selectedFileList)
```

Testing example

- Make a separate test file with known inputs and outputs
- You don't need to wait for the rest of the cell to run

```
In [ ]: def filelist_for_makam_and_form(fileList, makam, form):
            '''Selects/returns a file list (a subset of the whole list: 'fileList')
            given the filters: makam(str) and form(str)'''
            selectedFiles = []
            for filepath in fileList:
                filename = os.path.basename(filepath)
                if filename.split('--')[0] == makam and filename.split('--')[1] == filename:
                    selectedFiles.append(filepath)
            return selectedFiles

        # Let's pick two makams only differing in seyir and plot melodic curves
        # for first sections in the sazsemaisi form
        makams = ['rast', 'mahur']
        form = 'sazsemaisi'

        # Example
        txtFilesList = ['muhayyer--bozlak--serbest--havayi_da--asaf_guven.txt',
                        'ussak--sarki--nimsofyan--gordum_seni--sadettin_kaynak.txt',
                        'mahur--sazsemaisi--aksaksemai----nikolaki.txt',
                        'sipihr--pesrev--muhammes----dilhayat_kalfa.txt',
                        'mahur--sazsemaisi--aksaksemai--bahar_1--goksel_baktagir.txt',
                        'buselik--sarki--duyek--sevgim_bitti_mi--arif_sami_toker.txt']

        for index, makam in enumerate(makams):
            selectedFileList = filelist_for_makam_and_form(txtFilesList, makam, form)
            print('Files in makam {} and form {}'.format(makam, form))
            print(selectedFileList)
```

Documentation

- Great, you have some code. How does it work?
- Add a description of what the software does. What is the expected input format? Is there an example of what a successful run looks like?
- Is this the implementation of code which you published? After the paper is accepted, cite it!

File documentation

```
"""
This script processes the datafiles generated by the
foo-project download script

The input should be a json file, formatted like this:

{"itemid1": {"field1": "value1", "field2": False},
 "itemid2": {"field1": "value6", "field2": True}}

For each item key, a new file, `itemid.json` will be
created in the output directory containing the contents
of the value of this key.
"""
```

Project documentation

https://github.com/sertansenturk/ahenkidentifier

README.md

build passing | codecov 97% | code climate 4.0 | version 1.5.2 | DOI 10.5281/zenodo.259699 | License AGPL v3

ahenkidentifier

Identifies the ahenk (transposition) of a makam music recording given the tonic frequency and the symbol (or the makam)

If you are using ahenkidentifier in your work, please cite:

> Şentürk, S. (2016). Computational Analysis of Audio Recordings and Music Scores for the Description and Discovery of Ottoman-Turkish Makam Music. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain.

Usage

```
from ahenkidentifier.ahenkidentifier import AhenkIdentifier

ahenk = AhenkIdentifier.identify(tonic_freq, makam)
# or
ahenk = AhenkIdentifier.identify(tonic_freq, tonic_symbol)
```

The inputs are:

```
# tonic_freq        :   The frequency of the tonic in Hz.
# makam/tonic_symbol:   The algorithm can either accept the makam-slug or
#                                       the note symbol of the tonic in SymbTr format as a string (e.g. B4b1).
```

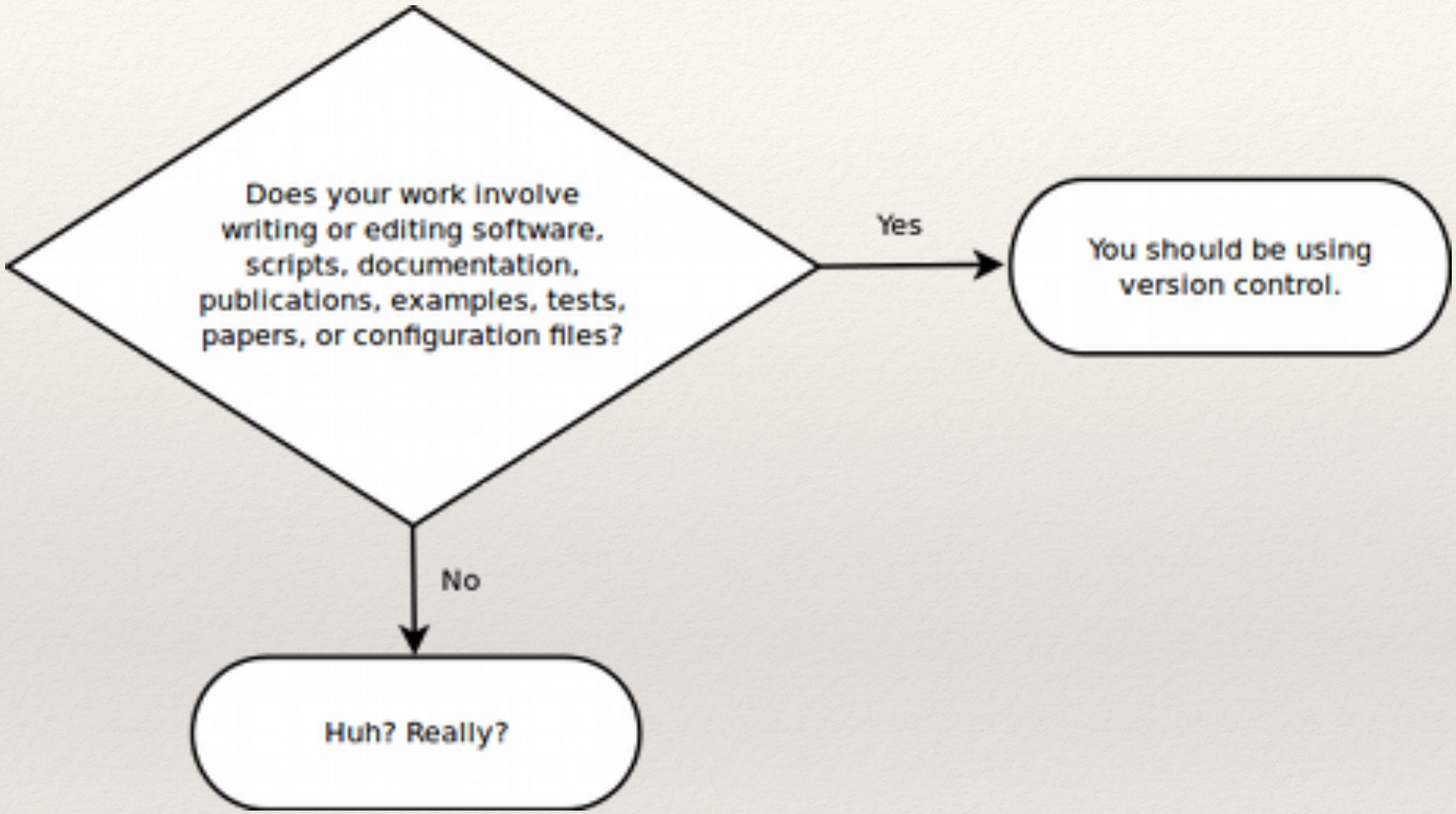
For the makam-slug names, check the json file in the data folder. The slugs are the same with the ones in the filenames of the scores in the SymbTr collection. The tonic symbols are notated as [Note pitch-class][Octave](Accidental Symbol(Holderian Comma)), e.g. B4b1

Why document?

- How does this save time?

- Because you are going to forget what your code does
  - Save yourself time having to read all of your files to work out how to run the software
  - If you want to publish the software, you now have documentation without any extra work

Source control



http://soundsoftware.ac.uk/why-version-control

Do you remember what you did last semester?

- Your code works
- You make a change
- You make another change
- …
- Copy your code to your laptop
- Make another change
- Your code doesn't work

- Easy way of keeping distributed backups (and knowing what is most up to date)

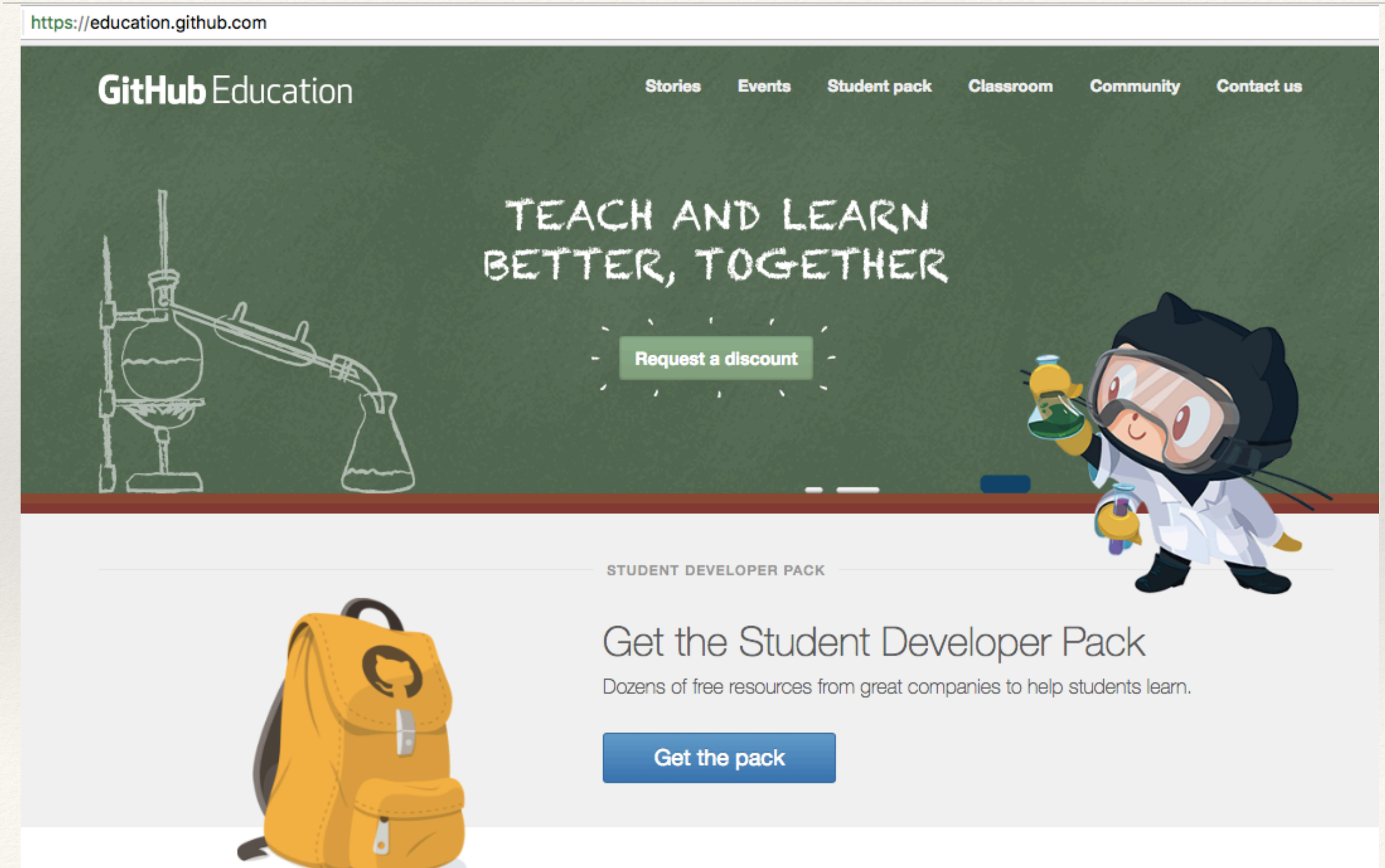
Source control

- Use git

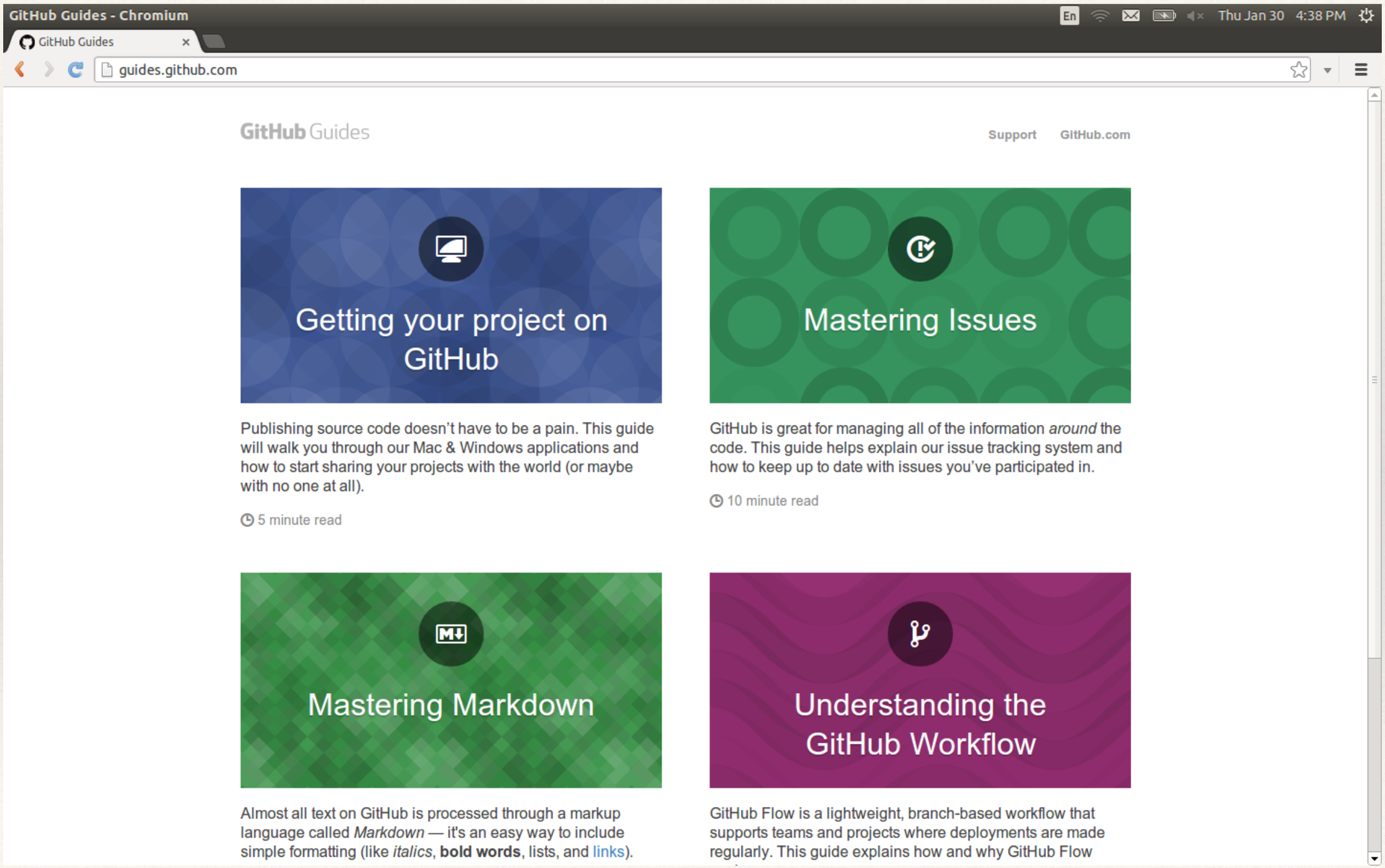
- Available for mac, windows, linux

- GitHub will give you a free place to store your code
- If you want to use mercurial or subversion, that's absolutely fine

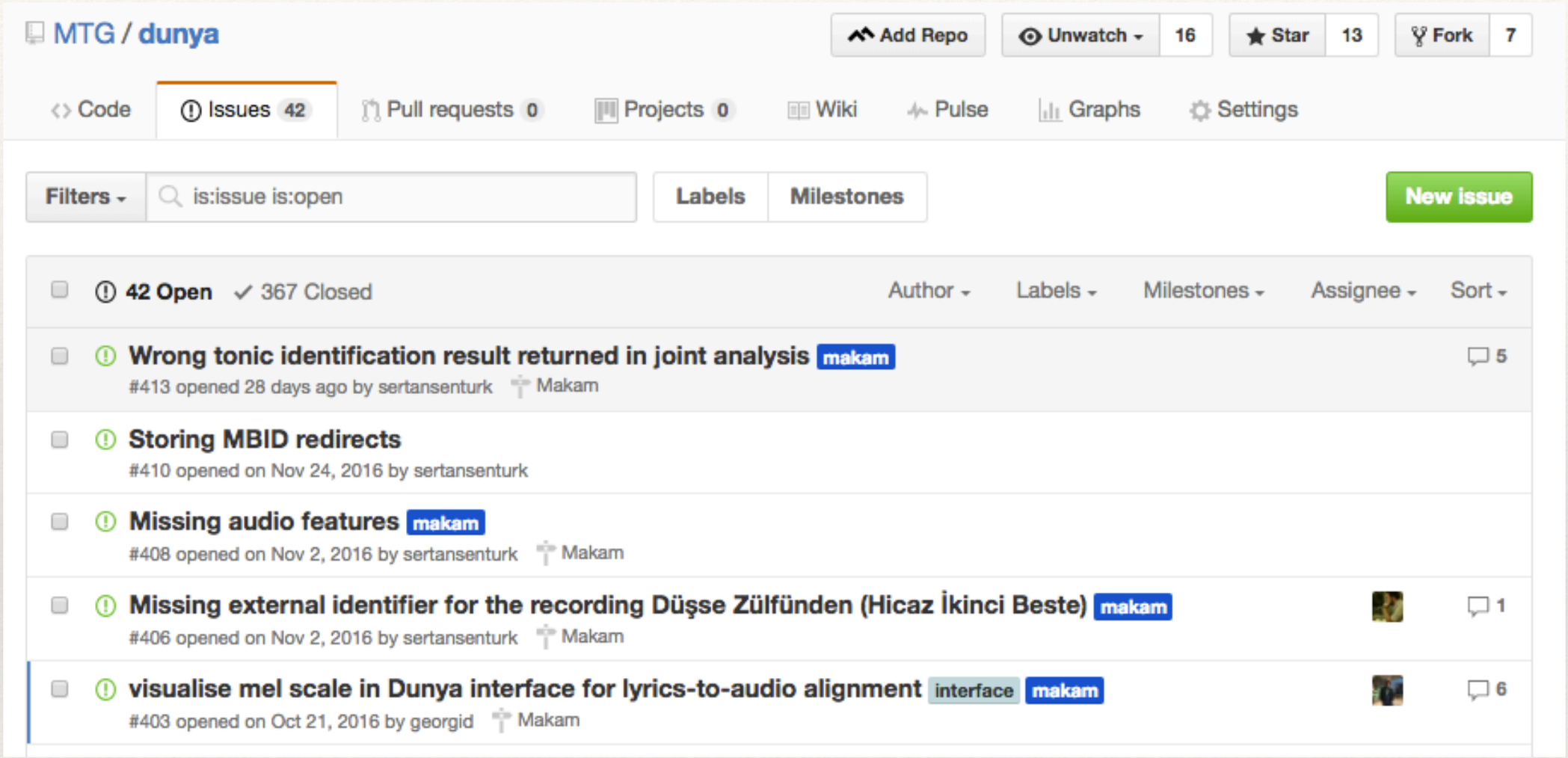
Academic Discount



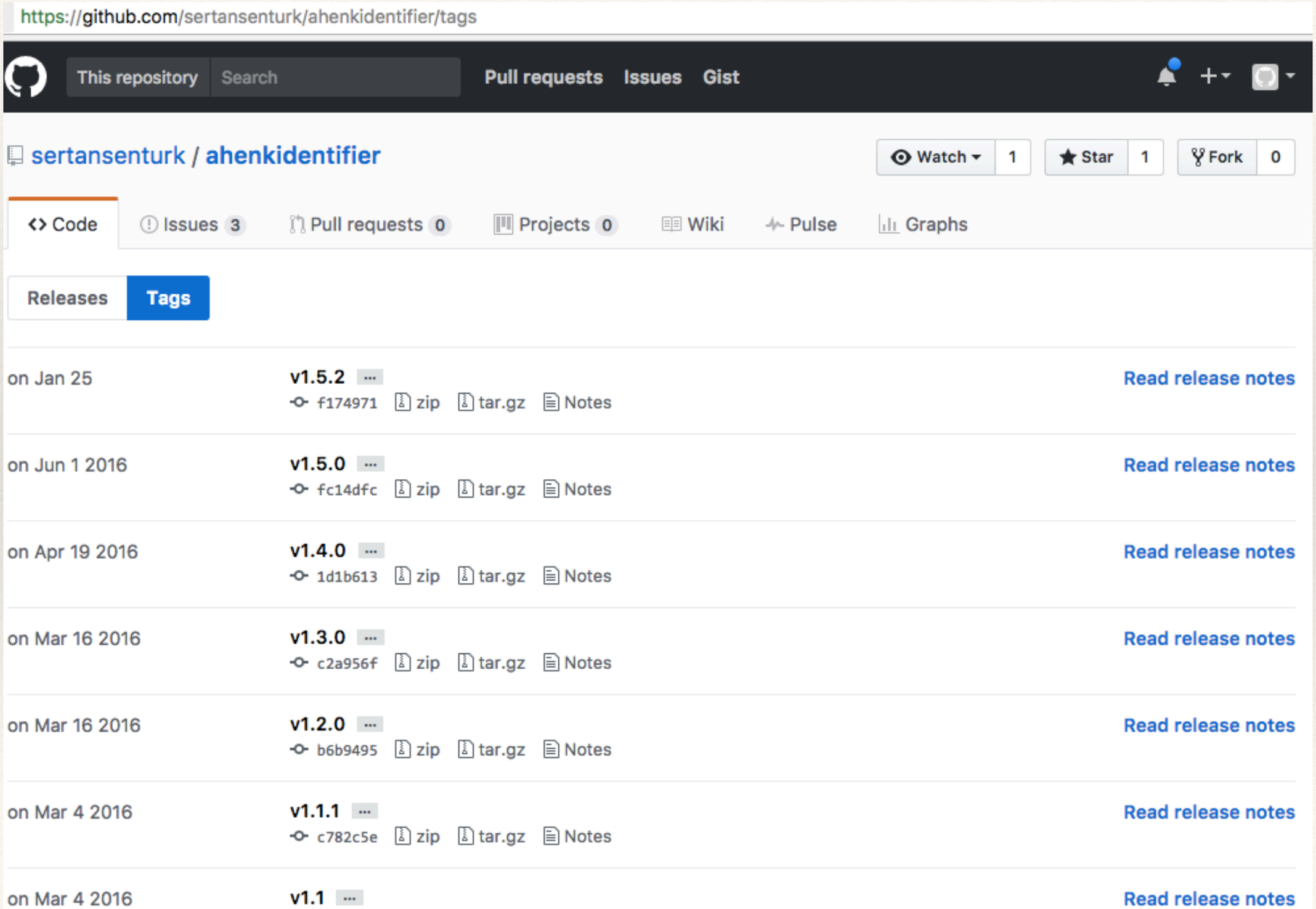
Tutorials


GitHub Guides - Chromium
Thu Jan 30 4:38 PM
GitHub Guides
guides.github.com
GitHub Guides
Support
GitHub.com
Getting your project on GitHub
Publishing source code doesn't have to be a pain. This guide will walk you through our Mac & Windows applications and how to start sharing your projects with the world (or maybe with no one at all).
5 minute read
Mastering Issues
GitHub is great for managing all of the information around the code. This guide helps explain our issue tracking system and how to keep up to date with issues you've participated in.
10 minute read
Mastering Markdown
Almost all text on GitHub is processed through a markup language called Markdown — it's an easy way to include simple formatting (like italics, bold words, lists, and links).
Understanding the GitHub Workflow
GitHub Flow is a lightweight, branch-based workflow that supports teams and projects where deployments are made regularly. This guide explains how and why GitHub Flow

Issues (Organise your life)



Keep track of versions



Keep track of versions

- Use git tags to mark the version that you used in a particular paper
- Do you want to try something new? make a branch and experiment without affecting your existing code

Data

- Your code probably processes data. If you're distributing code, can you also distribute data?
- This can be more difficult:
  - Is the data yours to distribute? copyright/personally identifying?
- Where do you host it?

Zenodo

- zenodo.org
- Designed for hosting public datasets
- Developed and maintained at CERN
- Hosted on the same storage infrastructure that stores data from the Large Hadron Collider
- Very generous storage limits (up to 50GB per dataset, email them if you need more!)

November 27, 2016

Software Open Access

MTG/SymbTr-pdf: SymbTr-pdf v2.4.3

Sertan Senturk

This release contains the music scores in the pdf format of the SymbTr collection v2.4.3:
https://github.com/MTG/SymbTr/tree/v2.4.3

Preview

SymbTr-pdf-v2.4.3.zip

! The previewer is not showing all the files

MTG-SymbTr-pdf-1e8b36d

.gitignore	10 Bytes
AUTHORS	122 Bytes
LICENSE	201 Bytes
README.md	787 Bytes
acem--ilahi--duyek--aldanma_dunya--zekai_dede.pdf	163.4 kB
acem--ilahi--nimevsat--calabim_bir--haci_bayram_veli.pdf	136.1 kB
acem--kupe--duyek--zulfunu--ahmet_avni_konuk.pdf	35.9 kB
acem--selam--devrikebir--asik-i_ger--huseyin_fahreddin_dede.pdf	193.1 kB
acem--seyir--sofyan--1--erol_bingol.pdf	29.0 kB
acem--seyir--sofyan--1--sefik_gurmeric.pdf	65.4 kB
acem--turku--duyek--ordunun_dereleri--ordu.pdf	139.2 kB
acemasiran--aranagme--agiraksak--1--.pdf	103.2 kB
acemasiran--aranagme--aksak--1--.pdf	99.1 kB
acemasiran--aranagme--curcuna--1--.pdf	102.3 kB
acemasiran--aranagme--semai--1--.pdf	100.7 kB
acemasiran--aranagme--senginsemai--1--.pdf	106.3 kB

Files (231.2 MB)

Name	Size	
MTG/SymbTr-pdf-v2.4.3.zip	231.2 MB	Preview Download

md5:5bb477412b7d6ed816e527c17f9b067b

Available in

GitHub

Publication date:
November 27, 2016

DOI:
DOI 10.5281/zenodo.174616

Related identifiers:
Supplement to:
https://github.com/MTG/SymbTr-pdf/tree/v2.4.3

License (for files):
Other (Open)

Share

Cite as

Sertan Senturk. (2016, November 27). MTG/SymbTr-pdf: SymbTr-pdf v2.4.3 (Version v2.4.3). Zenodo. http://doi.org/10.5281/zenodo.174616

Start typing a citation style...

Export

BibTeX CSL DataCite Dublin Core JSON JSON-LD MARCXML Mendeley

Zenodo + GitHib

- You can link a repository on GitHub to Zenodo so that every time you make a release you get a DOI to reference in your paper

Data Backups

- Is your data backed up?
- Thirty-five minutes spent in Langley's Willowbrook Shopping Centre cost a Surrey woman much more than she had anticipated. Langley RCMP say that while she was shopping from 1-1:35 p.m. last Monday, someone broke into her vehicle and stole a number of items, including a Mac iBook laptop containing the research she had compiled as she worked towards her PhD. "All that information was on that computer and she has no back-up file," said Langley RCMP spokesman Cpl. Brenda

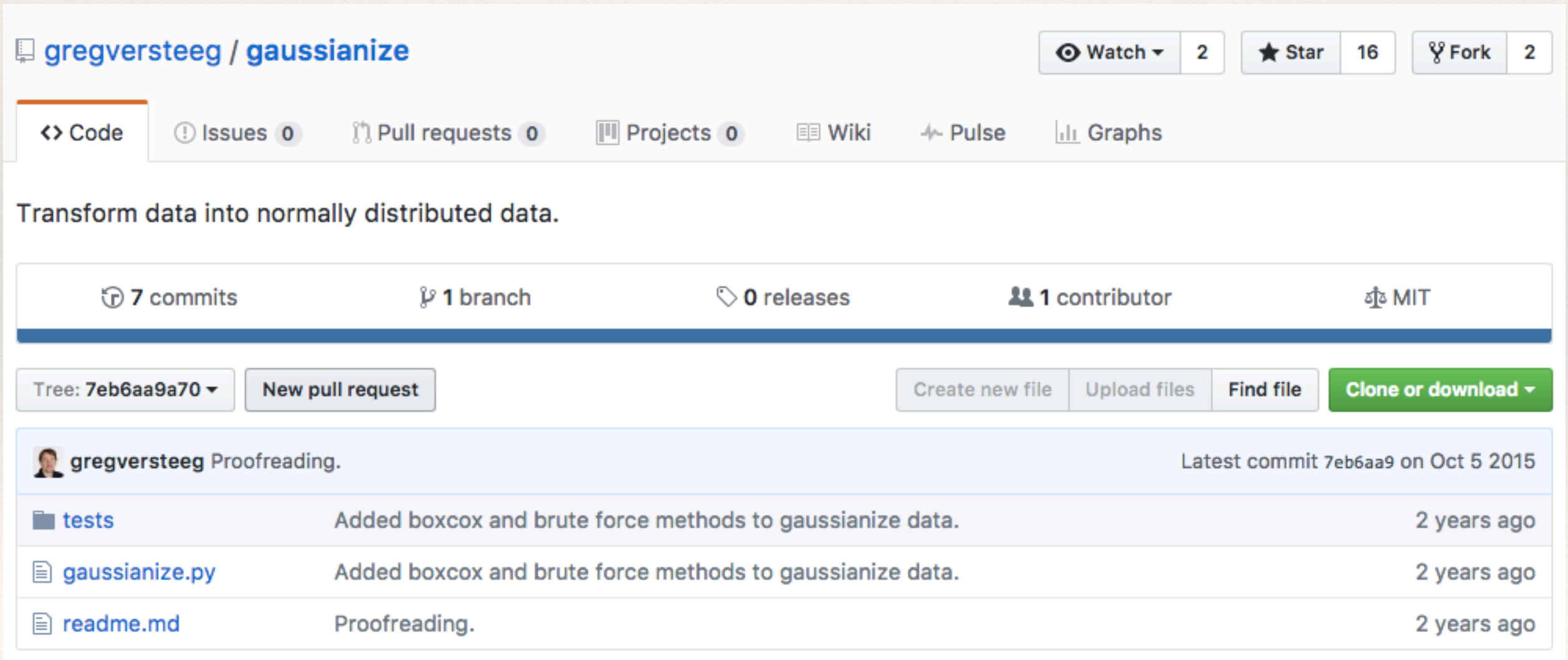
Plumbley et al. 2012. Research reproducibility workshop

Data Backups

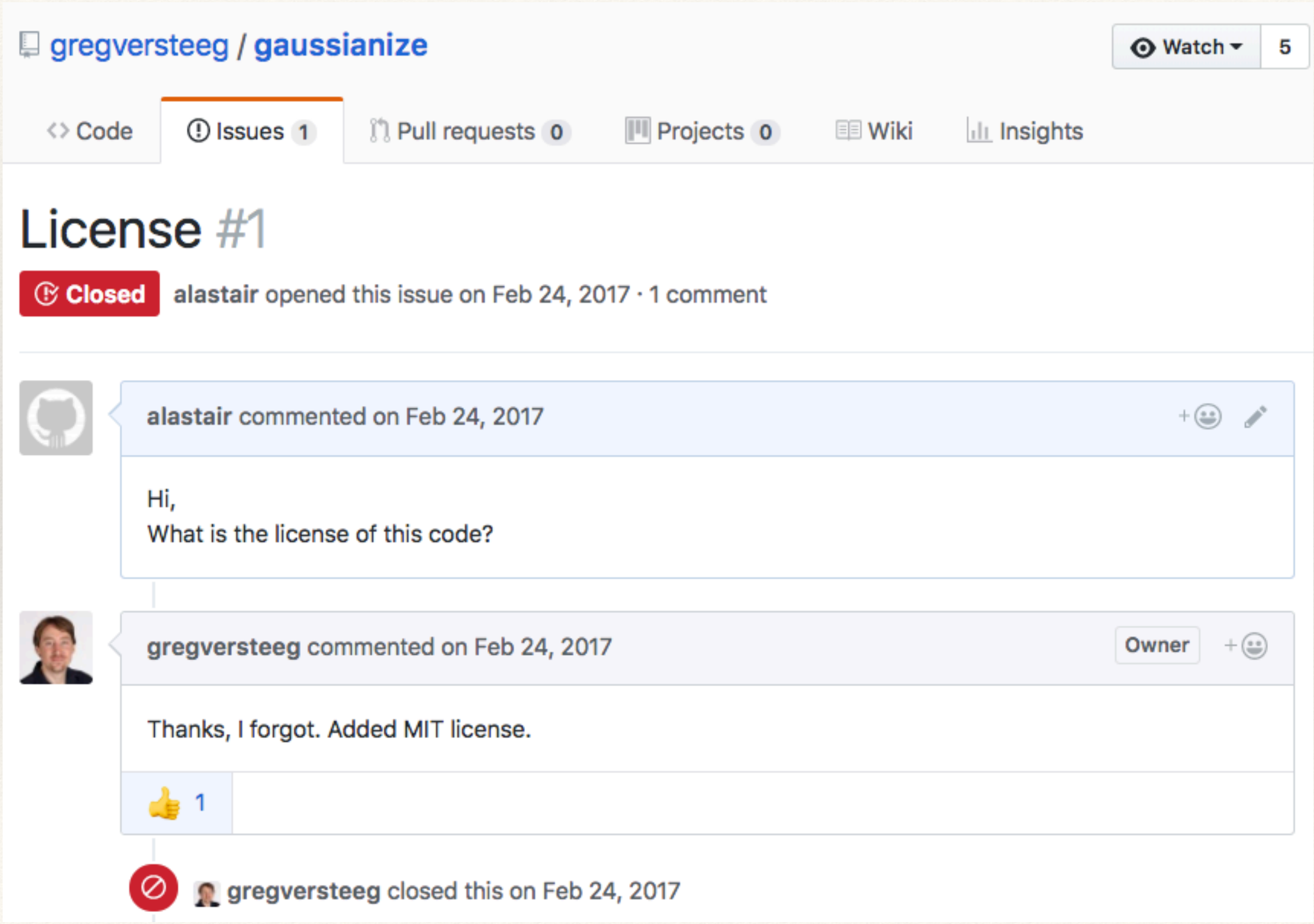
- Dropbox
- External hard drive
- Department servers/backup infrastructure

Licensing

- I found this code online which does something I need to do. I can download the code, so I can use it, right?



Licensing


gregversteeg / gaussianize
Watch 5
Code
Issues 1
Pull requests 0
Projects 0
Wiki
Insights
License #1
Closed
alastair opened this issue on Feb 24, 2017 · 1 comment
alastair commented on Feb 24, 2017
Hi,
What is the license of this code?
gregversteeg commented on Feb 24, 2017
Owner
Thanks, I forgot. Added MIT license.
1
gregversteeg closed this on Feb 24, 2017

Licensing

- No. Copyright law says that when you create something that no one can use or change it (without your permission)

gregversteeg / gaussianize

Watch 2 | Star 16 | Fork 2

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs

Transform data into normally distributed data.

8 commits | 1 branch | 0 releases | 1 contributor | MIT

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

gregversteeg committed on GitHub add license — Latest commit 23d0e46 19 days ago

tests	Added boxcox and brute force methods to gaussianize data.	2 years ago
LICENSE.txt	add license	19 days ago
gaussianize.py	Added boxcox and brute force methods to gaussianize data.	2 years ago
readme.md	Proofreading.	2 years ago

Open licenses

- Two main styles of open license:
  - GPL: "Use this code, but if you change it or integrate it into another system, your changes and that system also need to be available under this same open license"
  - BSD/MIT/Apache: "Do whatever you want. Really. Just say that I made it"
- Check with the copyright holder of your code (It might be the university)

How do you add a licence?

- Make a file called LICENCE or COPYING containing the contents of the licence
- GitHub will do this for you when you make a repository. It will ask you what licence you want and will automatically add the file

Reproducible research

- Sertan Şentürk (MTG PhD student) gave an example in a talk —3 weeks to reproduce an experiment with an external researcher (https://zenodo.org/record/255537)
- Do you know what the dependencies are? How to format the input data? How to execute the software? Does your code still work?
- This is not just for others. For you too
- "The first person to benefit by making your work reproducible is you!" - Sertan Şentürk

Reproducible research

- It's my belief that following these recommendations will make it easier to have research that can be verified and extended

- An entirely reproducible paper. Can someone else (you) run all of your experiments, create all tables, graphs, and generate your paper/thesis by running just 1 command? (Victoria Stodden, University of Illinois at Urbana-Champaign)

Wrapping up

- This is a lot of work. Do I have to do it?

- You're saving time
- You'll become a better programmer (easier to get a job)
- You'll actually finish your PhD on time!

Thanks!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowsa-Curie grant agreement No. 765068.