Make your code count: Leveraging open-source tools in quantum technology

Galileo Galilei Institute – Summer School on Quantum Computing and Sensing – June 22nd, 2021

Nathan Shammah
nathan@unitary.fund

Creating a quantum technology ecosystem
faster, better, and to benefit everyone.

Outline

- About Unitary Fund
  - Community activities
  - In-house research
- Quantum technology
- Open-source software (OSS)
  - An overview of scientific OSS
  - Make your code count

Non-profit helping create a quantum technology ecosystem faster, better, and to benefit everyone.

Our supporters

Our partners

https://unitary.fund/

Unitary Fund Team

Team

Will Zeng, PhD
President. Head of Quantum Research at Goldman Sachs. Fmr. product/sw lead at Rigetti. Oxford quantum algorithms PhD.

Sarah Kaiser, PhD
Community Manager. Co-founder of Q# Community, Microsoft MVP, activist for open source and diversity in quantum. UWaterloo PhD in quantum computing.

Ryan LaRose
NASA Fellowship PhD student at University of Michigan. Fmr at Alphabet X. Wrote first paper benchmarking quantum software packages.

Nathan Shammah, PhD
CTO. Lead developer at QuTiP. Visiting scientist at RIKEN. PhD in quantum physics from Univ. of Southampton.

Andrea Mari, PhD
> 40 peer reviewed scientific publications. Contributor to Pennylane. Fmr. researcher at Xanadu. Univ. of Postdam PhD in quantum information.

Dan Strano
Full stack web engineer. Lead developer on the qrack quantum simulator.

Vincent Russo, PhD
Post-quantum security developer at ISARA. Lead dev on toqito quantum info package. PhD from UWaterloo.

Advisory Board

15 volunteer experts in quantum systems & software from:

Community

60+ grantees

21 countries

Full-time Part-time

Unitary Fund Programs

Microgrant Program

Small microgrants => big impact

47 grants in 21 countries;
10+ publications;
30+ libraries with 1167 stars, 50+ contributors and ~6k commits
Helped 12 folks get into quantum tech full time

2 new startups, 1 new non-profit

>$100k

Granted to date

Unitary Labs

- Open source research team
- Targeting impactful and open niches that aren’t monetizable by companies

Mitiq: Error mitigating quantum compiler

```
from qiskit import QuantumCircuit
from mitiq import mitigate_executor
qskt_noisy_sim = mitigate_executor(qs)
circ = QuantumCircuit(1, 1)
for __ in range(120): circ.x(0)
circ.measure(0, 0)

expectation = qskt_noisy_sim(circ)
print(f"Error is {1 - expectation:.{3}}")
```

```
Error is 0.0582
```

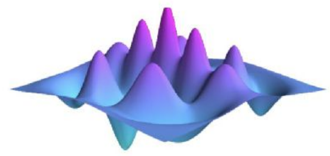
Built on Original Research:

Digital Zero Noise Extrapolation for Quantum Error Mitigation: https://arxiv.org/abs/2005.10921

Mitiq: A software package for error mitigation on noisy quantum computers: https://arxiv.org/abs/2009.04417

QuTiP: Quantum Toolbox in Python

Unitary Fund is setting up governance and fiscal sponsorship for a critical open source quantum technology package



qutip.org/

>30k annual downloads, >2500 citations

- Sep 2019 downloads 152k total

Unitary Fund

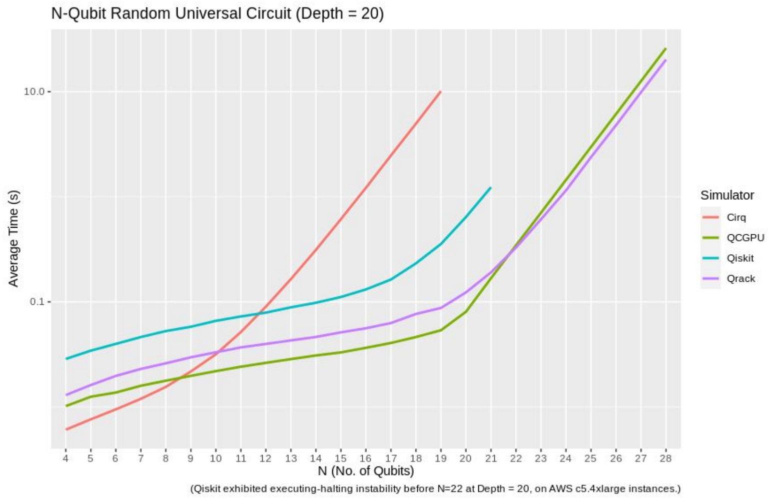
Microgrants: Building state of the art open software

QRack

SUPPORTED BY UNITARY FUND

an open source, comprehensive, GPU-accelerated framework for simulating universal quantum processors.

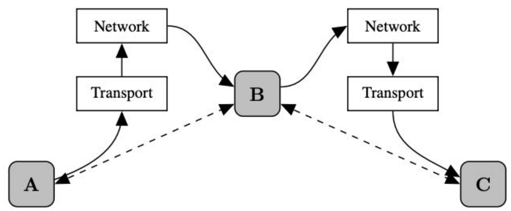
Better performance that industry options.



QuNetSim

SUPPORTED BY UNITARY FUND

To Stephen DiAdamo to develop the first full featured software stack for quantum network protocols.



OLSQ

SUPPORTED BY UNITARY FUND

To Daniel Tan to develop and open source the Optimal Layout Synthesizer for Quantum Computing, OLSQ. This compiler beats other benchmarks on optimal layout of computational qubits onto physical qubits.

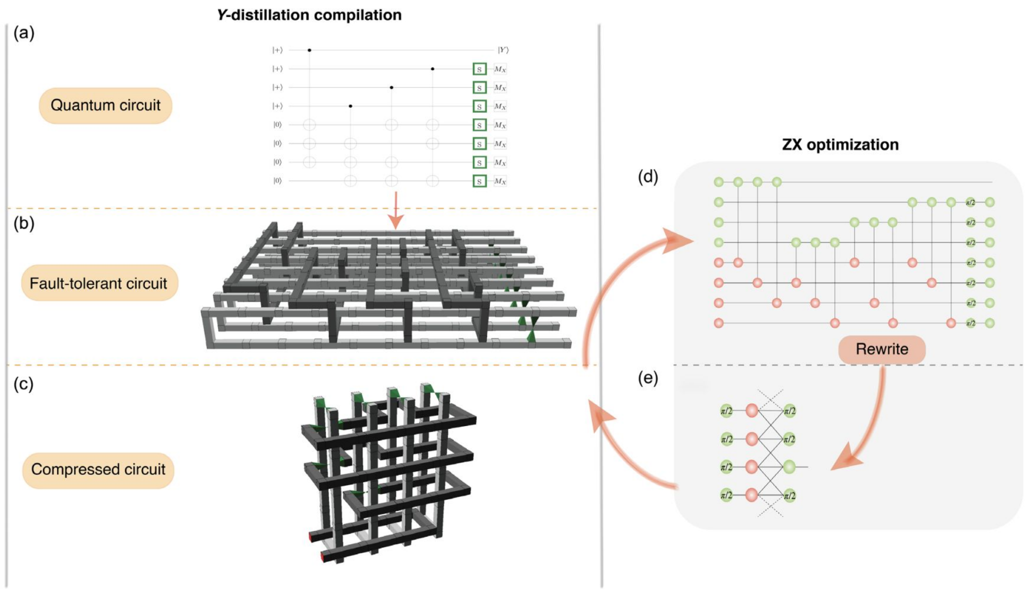
Table 4. Evaluation of QAOA-OLSQ

M	t|ket⟩ Depth	SWAP	TB-OLSQ Depth	SWAP	Depth Reduction	SWAP Reduction	QAOA-OLSQ Depth	SWAP	Depth Reduction	SWAP Reduction
10	16	7.3	6.9	7.3	56.7%	0	6.5	5.5	59.3%	23.6%
12	17.8	11.7	8.5	9.3	52.3%	20.4%	5.6	5.8	67.3%	46.2%
14	19.0	13.2	9.0	12.3	52.6%	6.8%	6.0	6.6	68.3%	48.0%
16	21.7	20.2	9.1	13.6	58.2%	32.7%	6.4	6.9	70.2%	62.6%
18	25.5	26.7	8.9	14.5	64.9%	45.7%	6.0	8.3	75.5%	65.7%
20	30.6	37.5	9.3	16.3	68.9%	57.7%	7.2	10.8	75.7%	68.8%
22	29.8	38.4	10.3	17.8	65.4%	53.6%	7.8	14.2	73.7%	61.8%
Geometric Mean					59.5%	29.4%			70.2%	53.8%

Open-source scientific software: research at scale

PyZX: compress quantum programs efficiently with ZX calculus

EFFECTIVE COMPRESSION OF QUANTUM BRAIDED CIRCUITS … PHYS. REV. X **10,** 041030 (2020)



aided tools. We found it helpful to put several of the larger circuits through an initial round of automated reduction with PyZX, an open source PYTHON library designed to reduce, validate, and visualize ZX-calculus diagrams [36]. PyZX applies a recursive, greedy algorithm [37]. Though the strategy of the PyZX library achieves significant reductions, it does not necessarily take into account the additional gate costs mentioned above (for instance, the reduced graphs of PyZX tend to have many Hadamard gates). Nonetheless, having reduced the overall graph size, it became feasible in isolation to tackle the $\pi/4$, $\pi/8$, and Hadamard gates by hand.

Effective Compression of Quantum Braided Circuits Aided by ZX-Calculus
Michael Hanks, Marta P. Estarellas, William J. Munro, and Kae Nemoto
Phys. Rev. X **10**, 041030 – Published 11 November 2020
https://journals.aps.org/prx/abstract/10.1103/PhysRevX.10.041030

Unitary Fund

Quantum Tech: Quantum computing

Open-source and quantum technologies

NATURE | COMMENT

Commercialize quantum technologies in five years

Masoud Mohseni, Peter Read, Hartmut Neven, Sergio Boixo, Vasil Denchev, Ryan Babbush, Austin Fowler, Vadim Smelyanskiy & John Martinis

03 March 2017

IBM adds new API to quantum computing cloud service

Posted yesterday by Ron Miller (@ron_miller)

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | Fo

Archive > Volume 549 > Issue 7671 > Comment > Article

NATURE | COMMENT

First quantum computers need smart software

Will Zeng, Blake Johnson, Robert Smith, Nick Rubin, Matt Reagor, Colm Ryan & Chad Rigetti

13 September 2017

Early devices must solve real-world problems, urge Will Zeng and colleagues.

PDF | Rights & Permissions

Subject terms: Quantum physics • Quantum information • Mathematics and computing •

A Recent Review:

Mark Fingerhuth, Tomáš Babej, and Peter Wittek,

Open source software in quantum computing,

PLoS ONE **13** (12): e0208561 (2019).

Unitary Fund

Quantum Tech: Open Source Libraries

More open-source is empowering broad research in the field

Library	Year	Creators	Institution	Language	Description
QuTiP Quantum Toolbox in Python	2012	Rob Johannson Paul Nation Franco Nori	RIKEN	**Python**	Simulation of open quantum systems; quantum optics, cavity QED.
QNet	2012	Nikolas Tezak, Michael Goerz Hideo Mabuchi	Stanford	**Python**	Computer algebra package for quantum mechanics and photonic quantum networks
QuantumOptics.jl	2017	Sebastian Krämer *et al.*	U Innsbruck IQOQI	Julia	Quantum optics and open quantum systems framework inspired by the QO toolbox in Matlab and QuTiP
ProjectQ	2016	Damian S. Steiger Thomas Häner Matthias Troyer	ETH Zurich	**Python**	Hardware-agnostic framework with compiler and simulator with emulation capabilities.
OpenFermion	2017	Ryan Babbush *et al.*	Google (unofficial)	**Python**	Fermionic potential calculations for quantum chemistry
NetKet	2018	Giuseppe Carleo	The Simons Foundation	C++ **Python**	Studying many-body quantum systems with artificial neural networks and ML techniques.
STRAWBERRY FIELDS	2018	Nathan Killoran *et al.*	Xanadu Inc	**Python**	Photonic quantum computing with continuous-variable optical circuits

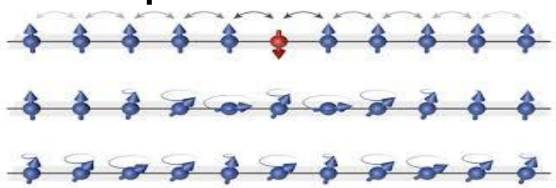
Checkout more open-source projects at https://qosf.github.io

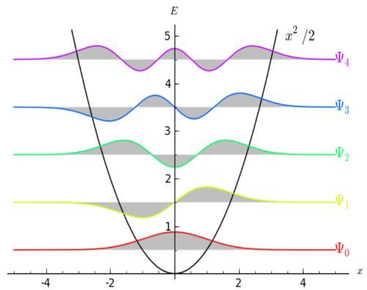
Unitary Fund

QuTiP: The Quantum *Physics* Simulator

The **Qu**antum **T**oolbox **i**n **P**ython

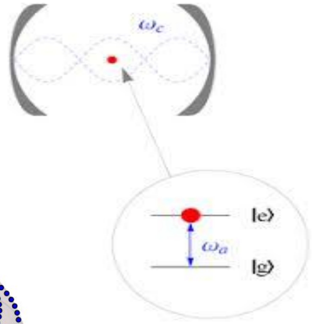
Spin Lattices

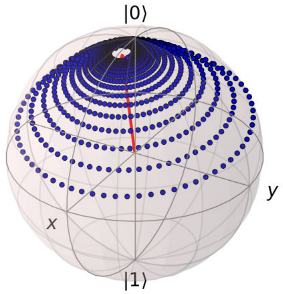




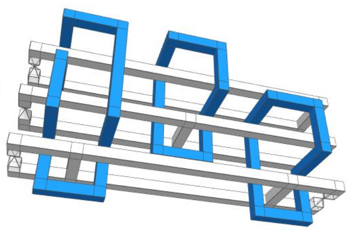
Condensed Matter

Cavity QED

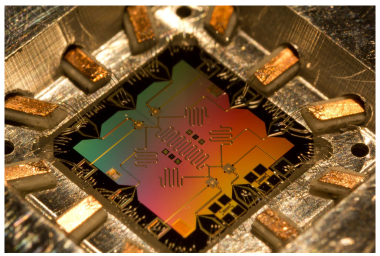




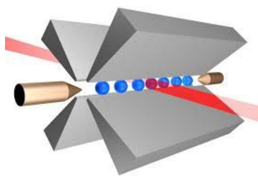
Quantum Optics



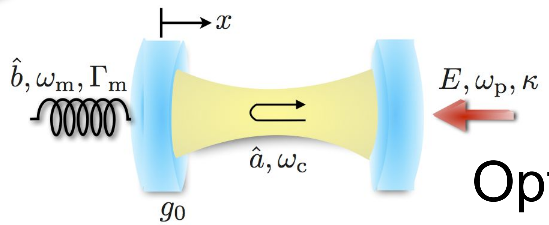
Quantum Error Correction



Superconducting Circuits



Ion Traps



Optomechanics

Unitary Fund

Open source for software, knowledge, research

Linux

WIKIPEDIA
The Free Encyclopedia

arXiv.org

THE LINUX FOUNDATION

WIKIMEDIA FOUNDATION

Cornell University

"$5Bn in contributed development costs"*
*linuxfoundation.org/press-release/2015/09/

Unitary Fund

Open source for software, knowledge, research

“Windows and MacOS are products, contrived by engineers
in the service of specific companies.

Linux/Unix, by contrast, is not so much a product
as it is **a painstakingly compiled oral history of the hacker subculture**.

It is our Gilgamesh epic.”

– Neal Stephenson

Unitary Fund

Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

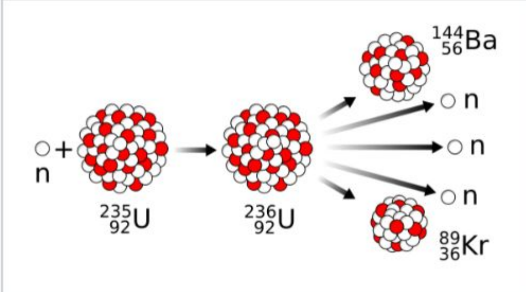
Contribute

Help
Learn to edit
Community portal
Recent changes
Upload file

Discovery of nuclear fission

From Wikipedia, the free encyclopedia

Nuclear fission was discovered in December 1938 by chemists Otto Hahn and Fritz Strassmann and physicists Lise Meitner and Otto Robert Frisch. Fission is a nuclear reaction or radioactive decay process in which the nucleus of an atom splits into two or more smaller, lighter nuclei. The fission process often produces gamma rays and releases a very large amount of energy, even by the energetic standards of radioactive decay. Scientists already knew about alpha decay and beta decay, but fission assumed great importance because the discovery that a nuclear chain reaction was possible led to the development of nuclear power and nuclear weapons.



The nuclear reaction theorised by Meitner and Frisch.

Hahn and Strassmann at the Kaiser Wilhelm Institute for Chemistry in Berlin bombarded uranium with slow neutrons and discovered that barium had been produced. They reported their findings by mail to Meitner in Sweden, who a few months earlier had fled Nazi Germany. Meitner and her nephew Frisch theorised, and then proved, that the uranium nucleus had been split and published their findings in *Nature*. Meitner calculated that the energy released by each disintegration was approximately 200 megaelectronvolts, and Frisch observed this. By analogy with the division of biological cells, he named the process "fission". Hahn was awarded the 1944 Nobel Prize in Chemistry for the discovery.

Unitary Fund

Discovery

Q: "Bohr's liquid drop model had not yet been formulated, so there was no theoretical way to calculate whether it was physically possible for the uranium atoms to break into large pieces." The model was introduced in the previous section as Gamov's liquid drop model. When did Bohr formulate his model?

A: The previous section said: The current model of the nucleus was the liquid drop model first proposed by George Gamov in 1930. His simple and elegant model was refined and developed by Carl Friedrich von Weizsäcker and, after the discovery of the neutron, by Werner Heisenberg and Niels Bohr Hawkeye7 (discuss) 03:28, 7 August 2020 (UTC)

Still, it kind of appears out of the blue because we, at that point, don't know when Bohr developed his own model. How about "and Niels Bohr (in 193x)."

Added "in 1936". Hawkeye7 (discuss) 05:57, 7 August 2020 (UTC)

Q: Last two paragraphs of "Objections" (why this title?) don't seem too relevant, should they be rather be in their respective biographies? The story is interesting, for sure, but the article is already too long.

A: It tells the reader how this team came together, and why Meitner and Hahn were engaged in digging in to this mystery instead of running the KWI for Chemistry. Meitner's later departure is also foreshadowed. (The title refers to objections to Fermi's claim to have discovered Hawkeye7 (discuss) 03:28, 7 August 2020 (UTC)

It's a biographic digression that I found a bit too long. Interesting, but long. No reason to fail GAN, though.

Q: Last few sentences in "Eureka" - it's unclear if all these dates were in 1939. Frisch conducted exp. in February, but mailed papers (about what?) in January.

A: Added "1939". Hawkeye7 (discuss) 05:57, 7 August 2020 (UTC)

Talk:Discovery of nuclear fission: Revision history

Help

View logs for this page (view filter log)

Filter revisions

External tools: Find addition/removal · Find edits by user (Alternate) · Page statistics · Pageviews · Fix dead links

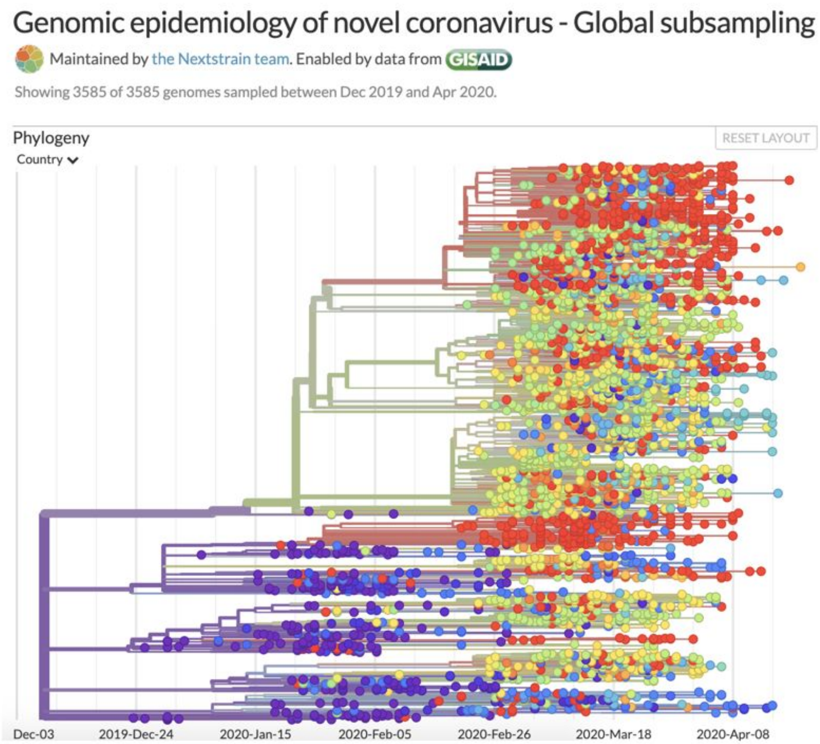
For any version listed below, click on its date to view it. For more help, see Help:Page history and Help:Edit summary. (cur) = difference from current version, (prev) = difference from preceding version, **m** = minor edit, → = section edit, ← = automatic edit summary

Compare selected revisions

- (cur | prev) 15:54, 22 June 2021 Philoserf (talk | contribs) . . (5,763 bytes) **(−622)** . . *(OneClickArchiver archived A Commons file used on this page or its Wikidata item has been nominated for deletion to Talk:Discovery of nuclear fission/Archive 1)* (undo)
- (cur | prev) 15:54, 22 June 2021 Philoserf (talk | contribs) . . (6,385 bytes) **(−956)** . . *(OneClickArchiver archived Commons files used on this page or its Wikidata item have been nominated for deletion to Talk:Discovery of nuclear fission/Archive 1)* (undo)
- (cur | prev) 15:54, 22 June 2021 Philoserf (talk | contribs) . . (7,341 bytes) (+1) . . *(Assessment: banner shell (Rater))* (undo)
- (cur | prev) 15:13, 14 May 2021 Wehwalt (talk | contribs) . . (7,340 bytes) (+23) . . *(Select as TFA for 22 June 2021)* (undo)
- (cur | prev) 17:01, 16 April 2021 Community Tech bot (talk | contribs) . . (7,317 bytes) **(+622)** . . *(Files used on this page or its Wikidata item are up for deletion)* (undo)
- (cur | prev) 18:46, 18 February 2021 BattyBot (talk | contribs) . . (6,695 bytes) (+34) . . *(→top: Added Template:WikiProject banner shell and other General fixes)* (undo) *(Tag: AWB)*
- (cur | prev) 00:33, 17 November 2020 Morgan695 (talk | contribs) . . (6,661 bytes) (+11) . . (undo)

Open source for open science: COVID-19

An aligned vision



https://nextstrain.org/ncov/global

Open Source

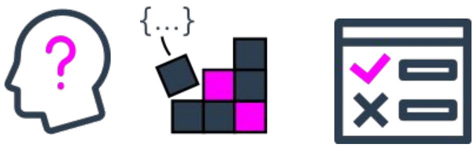
A new era for open source


Open-Source Basics
Read. Download. Deploy.
Definition: You can **read** the source code (open-source ≠ free).
Examples of open-source: Linux, Android, FireFox, MySQL, LibreOffice, **Python**.
Open-source deployment is **accelerating** many end-industries applications.

Open Source

A new era for open source

Open-Source Basics
Read. Download. Deploy.
Definition: You can **read** the source code (open-source ≠ free).
Examples of open-source: Linux, Android, FireFox, MySQL, LibreOffice, **Python**.
Open-source deployment is **accelerating** many end-industries applications.



Features for Developers
Learn. Debug. Deploy.
You can **learn** by reading the code and become a better developer.
You can edit the source code (licenses apply) and **collaborate** to existing projects.
You can submit **fixes** to bugs, propose improvements.

Unitary Fund

Open Source

A new era for open source

Open-Source Basics
Read. Download. Deploy.
Definition: You can **read** the source code (open-source ≠ free).
Examples of open-source: Linux, Android, FireFox, MySQL, LibreOffice, **Python**.
Open-source deployment is **accelerating** many end-industries applications.

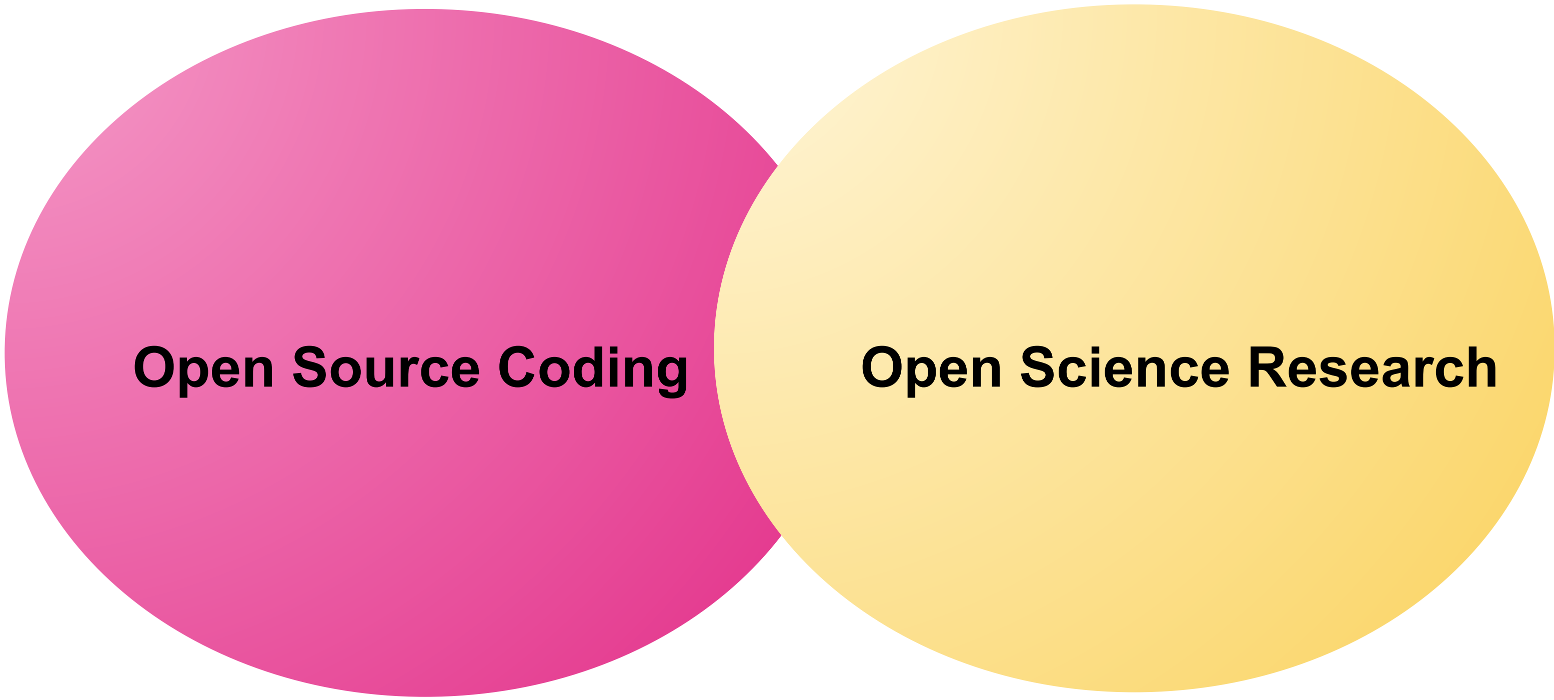
Features for Developers
Learn. Debug. Deploy.
You can **learn** by reading the code and become a better developer.
You can edit the source code (licenses apply) and **collaborate** to existing projects.
You can submit **fixes** to bugs, propose improvements.

Open-source for Businesses
Flexible. Valuable. Growing fast.
Source is generally free but companies can charge for additional services.
GitHub acquired for $7.5 bln (2018).GitLab: $100 mln in funding (2018).
Red Hat acquired by IBM for $34 bln (2019).
Machine learning ('AI') is driving **fast**, pervasive adoption of open-source libraries.

Unitary Fund

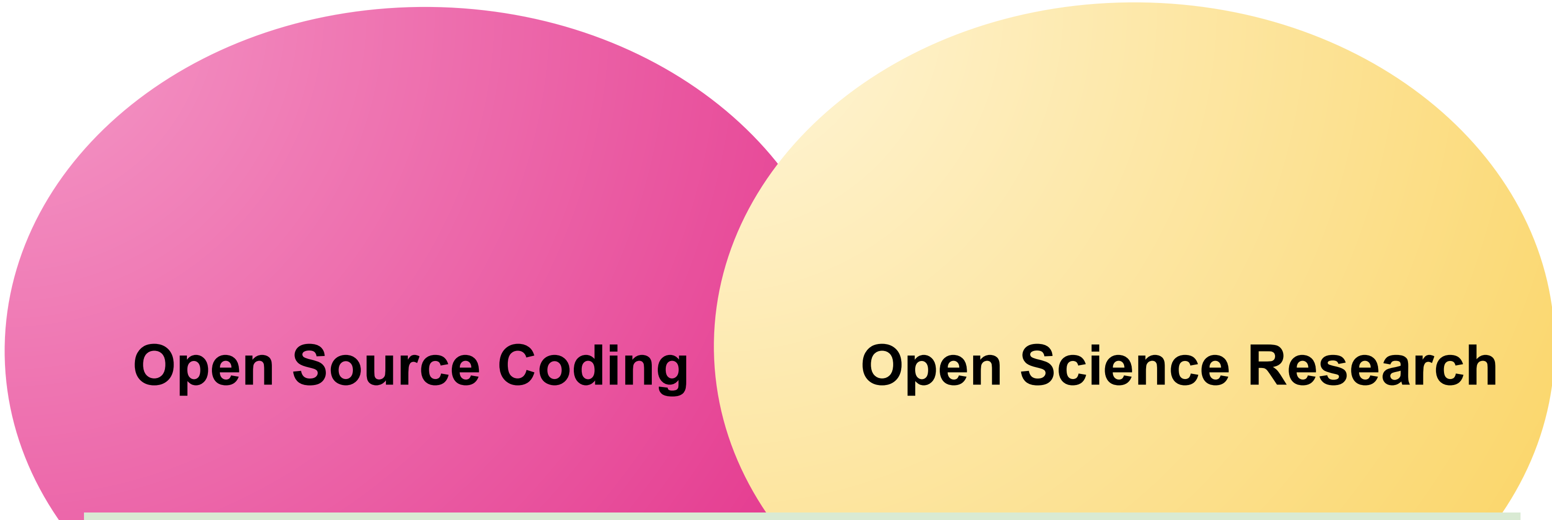
Open source and open science

Aligned vision



Open source and open science

Aligned vision

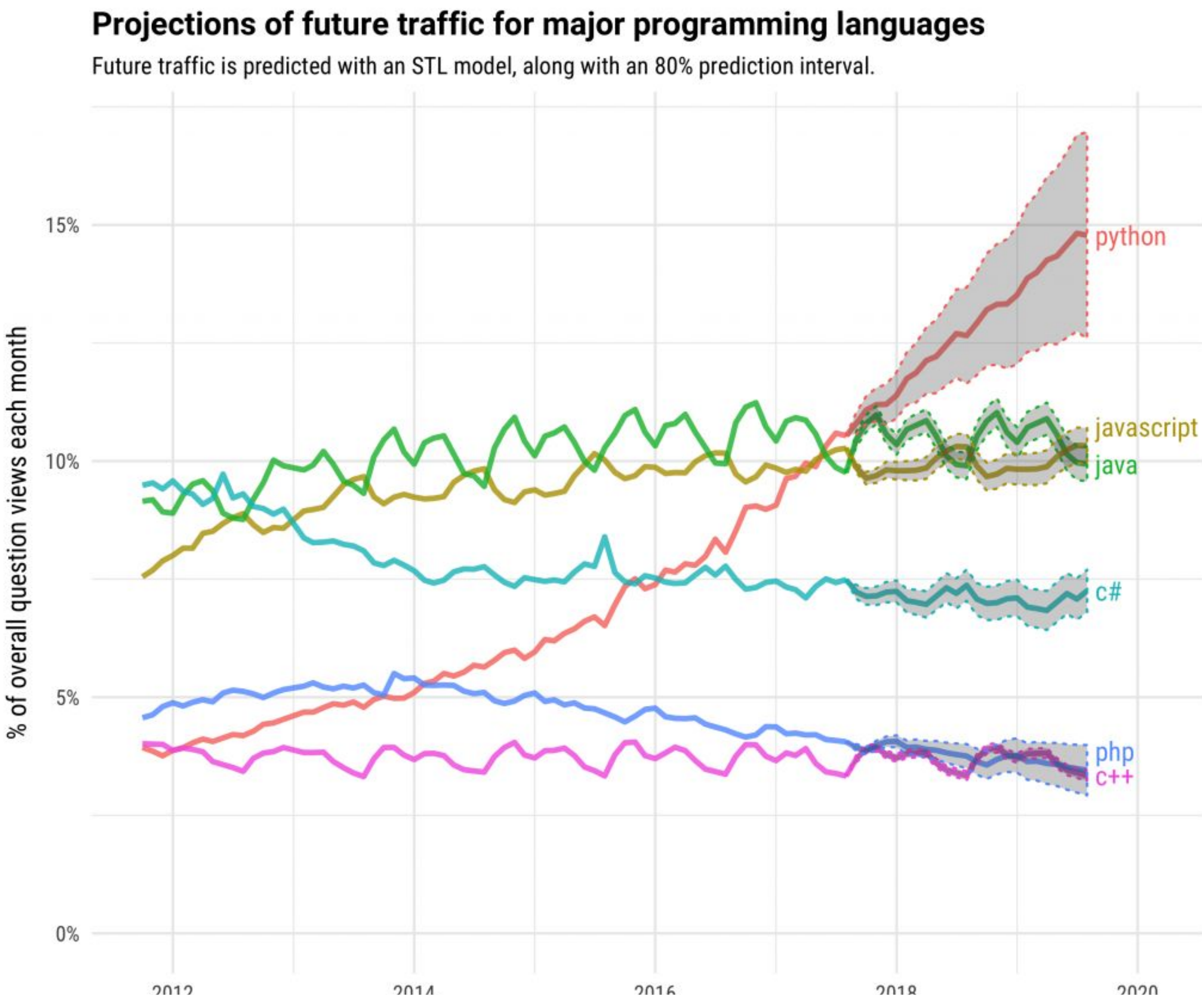


- Allow **access** to the research/project results, sharing knowledge.
- **Collaboratively** advance the field, building upon others' results.
- Coordinate large and delocalized **teams** working remotely.
- Make supporting data and code available for fast **reproducibility**.

Unitary Fund

The steady growth of Python

Empowered by a large open-source ecosystem



Source: David Robinson

Python’s strengths

A community-based programming language

Community

Resources

Tools

Python’s strengths

A new community-based programming language

Community

Python’s strengths

A modular architecture for well-maintained libraries

Resources

Python’s strengths

Interactive Jupyter notebooks run an IDE in your browser

Tools

Below we give basic examples on the use of `qutip.piqs`. In the first example the incoherent emission of N driven TLSs is considered. In tl
two-level system ensemble is a subsystem coupled to another subsystem, a bosonic cavity. Similar considerations apply to the coupling to
subsystems (a single qubit, another two-level system ensmble).

```
In [1]: import matplotlib.pyplot as plt
        import matplotlib as mpl
        from matplotlib import cm

        from qutip import *
        from qutip.piqs import *

        import matplotlib.animation as animation
        from IPython.display import HTML
        from IPython.core.display import Image, display
```

1. N Qubits Dynamics

We study a driven ensemble of N TLSs emitting incoherently,

$$H_{\text{TLS}} = \hbar\omega_0 J_z + \hbar\omega_x J_x$$

$$\dot{\rho} = \mathcal{D}_{\text{TLS}}(\rho) = -\frac{i}{\hbar}[H_{\text{TLS}}, \rho] + \sum_{n=1}^{N} \frac{\gamma_{\text{E}}}{2} \mathcal{L}_{J_{-,n}}[\rho]$$

Open Source

From reproducible data to reusable code.

gael-varoquaux.info

G. Varoquaux: Group leader at INRIA in Paris.
One of the **scikit-learn** core developers.

Open Source

From reproducible data to reusable code.

Beyond Reproducibility

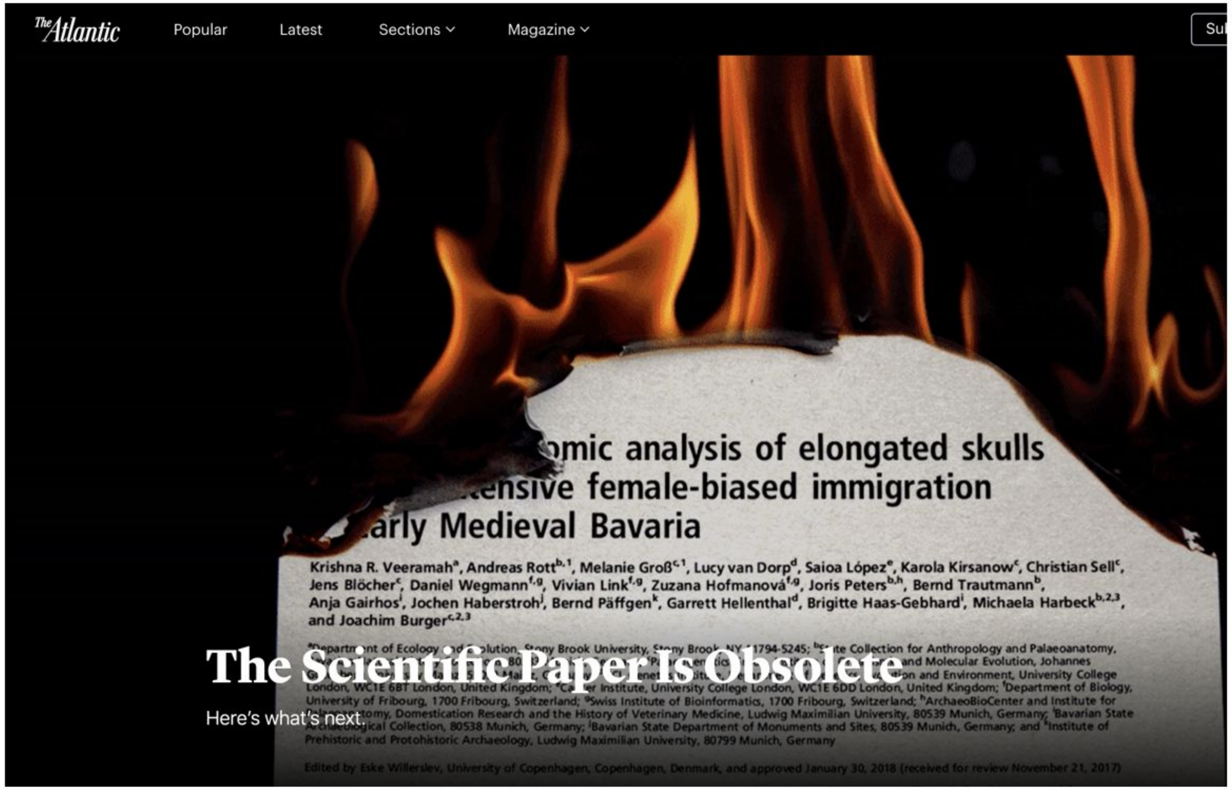
		DATA	
		Same	Different
ANALYSIS	Same	Reproducible	Replicable
	Different	Robust	Generalizable

https://dx.doi.org/10.6084/m9.figshare.7140050

x

Reproducibility in Science

A bold stance…



https://www.theatlantic.com/science/archive/2018/04/the-scientific-paper-is-obsolete/556676/

Reproducibility: Online

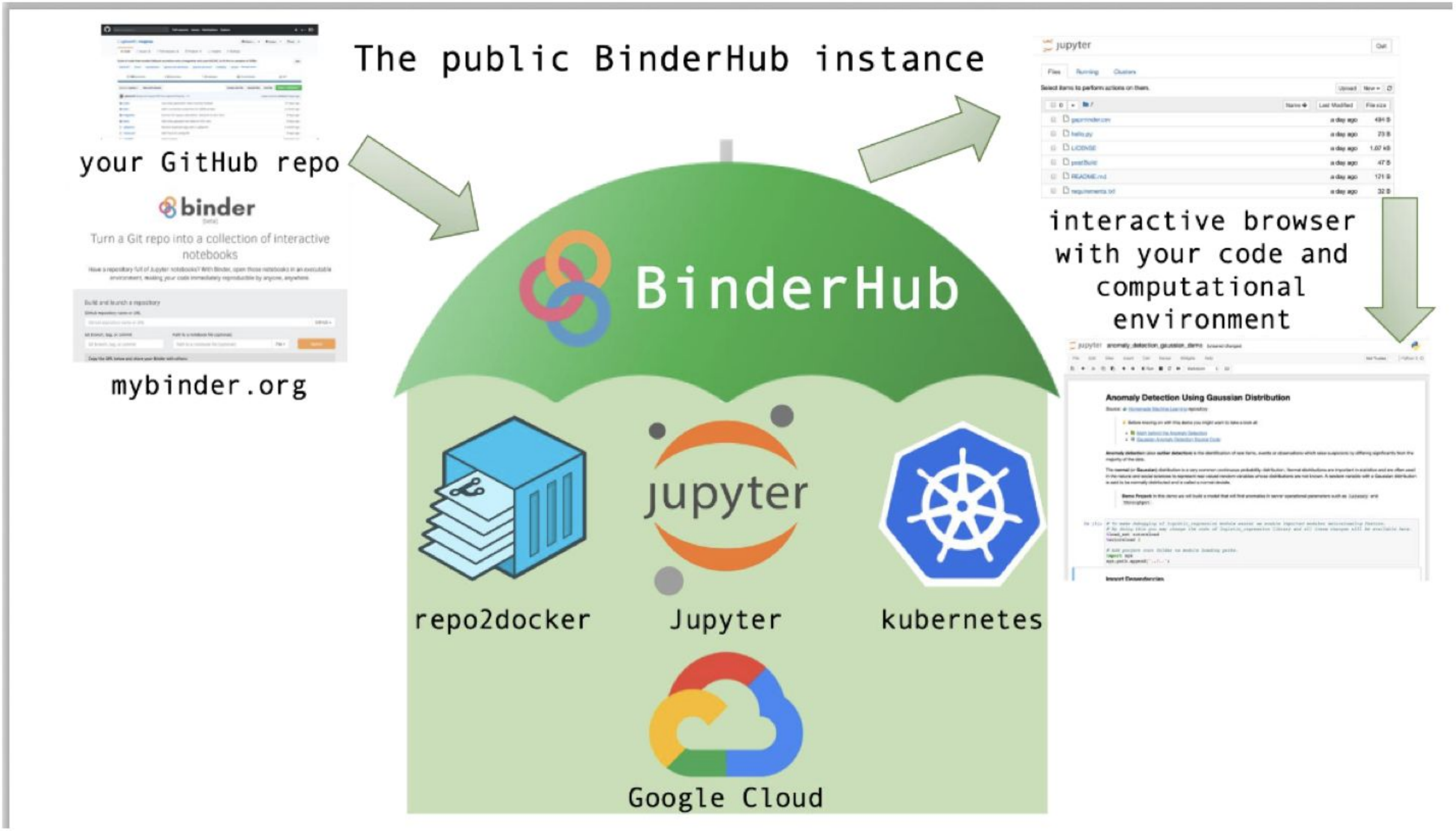


Figure from @TuringInst: Kirstie Whitaker, Why you need a reproducible computing environment

https://zenodo.org/record/2598530#.XQDf8tP7Q5g

X

Make your code count

A Guide to building your open-source scientific computing project in Python.

SUPPORTED BY UNITARY FUND

A cheatsheet to develop a scientific open-source library from scratch.

0: Open
1: Code
2: Develop

3: Test
4: Pack
5: Document

6: Distribute
7: Publish
8: Host

Open, Code, Develop

Test, Pack, Document

Distribute, Publish, Host

https://github.com/nathanshammah/make-your-code-count

Make your code count

A Guide to building your open-source scientific computing project in Python.

SUPPORTED BY UNITARY FUND

A cheatsheet to develop a scientific open-source library from scratch.

Shahnawaz Ahmed
Chalmers, Sweden
(RIKEN, Japan)

0: Open
1: Code
2: Develop

3: Test
4: Pack
5: Document

6: Distribute
7: Publish
8: Host

Open, Code, Develop

python git
GitLab GitHub

Test, Pack, Document

Travis CI
SPHINX Python Documentation Generator
Read the Docs

Distribute, Publish, Host

python Package Index CONDA
zenodo binder

https://github.com/nathanshammah/make-your-code-count

Open Science through Open Source: Making code count

The tools of open source

Code & Testing		Documentation		Publication	
GitHub GitLab	Create code collaboratively; host it and perform version control	SPHINX Python Documentation Generator	Self-generate a documentation for the library from commented functions	CONDA python Package Index	Install easily software on multiple platforms, ensuring update compatibilities
Travis CI	Enstrengthen code with independent testing of functions, from the cloud	Read the Docs	Freely host a library documentation with dedicated markup options	zenodo	Keep data safe and track release control, with immediate DOI for bibliographic records

An **OpenAIRE** + CERN project

x

Unitary Fund

Python: Function, Class, Objects

piqs/dicke.py

```
def num_dicke_states(N):
    """
    Calculate the number of Dicke states.

    Parameters
    ----------
    N: int
        The number of two-level systems.

    Returns
    -------
    nds: int
        The number of Dicke states.
    """
    nds = (N/2 + 1)**2 - (N % 2)/4
    return int(nds)
```

```
>> from piqs import num_dicke_states
>> print(num_dicke_states(30))
256
```

```
class Dicke(object):
  """
  A Dicke class defining an ensemble of spins

  Parameters
  ----------
  N: int
    Number of two level systems
  """
  def __init__(self, N, H=None, emission=0.):
          self.N = N
          self.H =H
          self.emission = emission

  def liouvillian(self):
    """
    Calculate the Liouvillian superoperator.
    ......
```

```
>> from piqs import Dicke
>> ensemble = Dicke(30, emission=0.1)
>> L = ensemble.liouvillian()
```


Unitary Fund

Git and GitHub

- Version control
- Collaborative code
- Online and open-source

paper.tex
paper_suggestions2.tex
paper2.tex
paper_deleted_sec1.tex
paper_paul_details.tex
paper_final.tex
paper_final2.tex

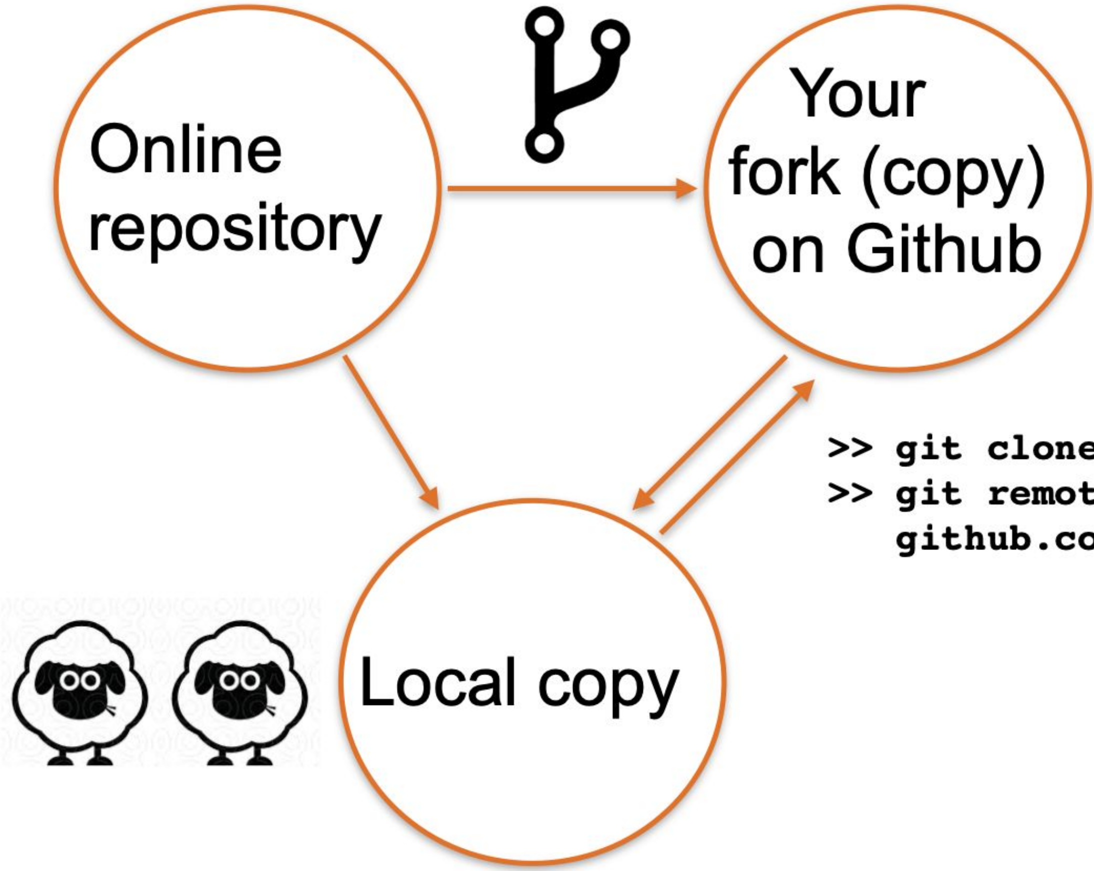
xkcd

x

Git: Setup

`github.com/qutip/qutip`

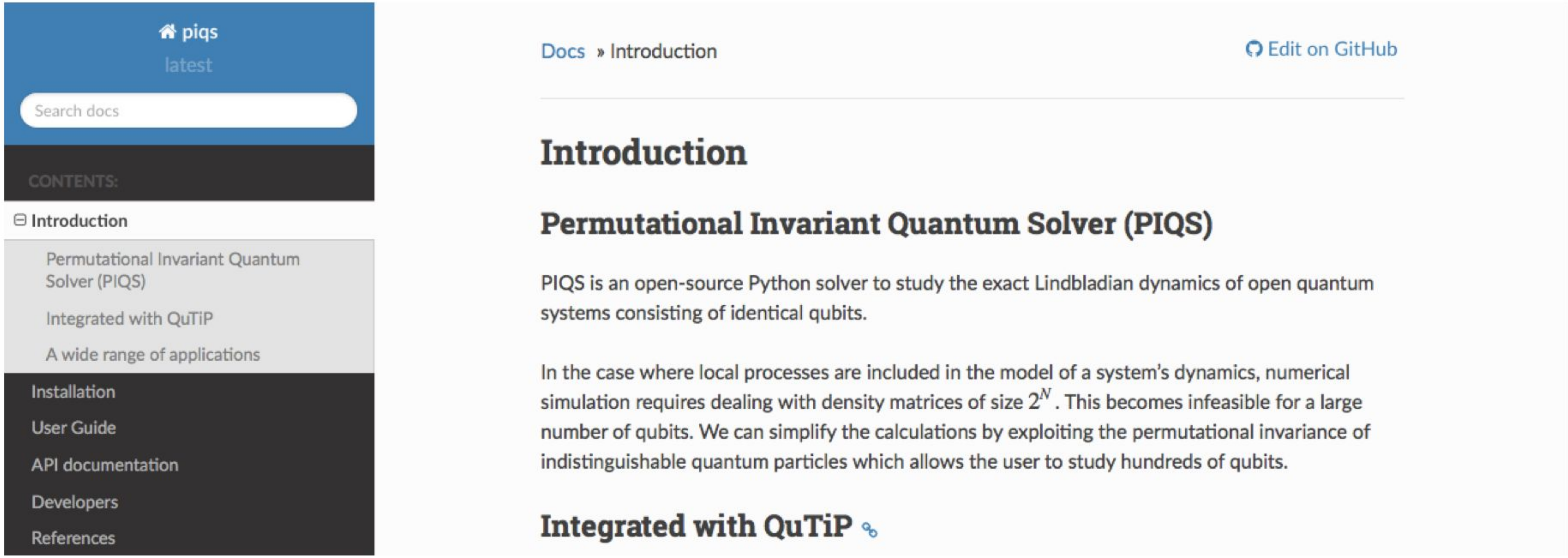
`github.com/paul/qutip`




```
>> git clone github.com/paul/qutip
>> git remote add upstream
   github.com/qutip/qutip
```

Documentation

PDF LaTeX and more …



Documentation

Auto generate with Sphinx

```
>> sphinx-quickstart
|-- doc/
     |-- MakeFile
     |-- make.bat
     |-- source
          |-- conf.py
          |-- conf.py
          |-- index.rst
          |-- intro.rst
```

Edit configurations

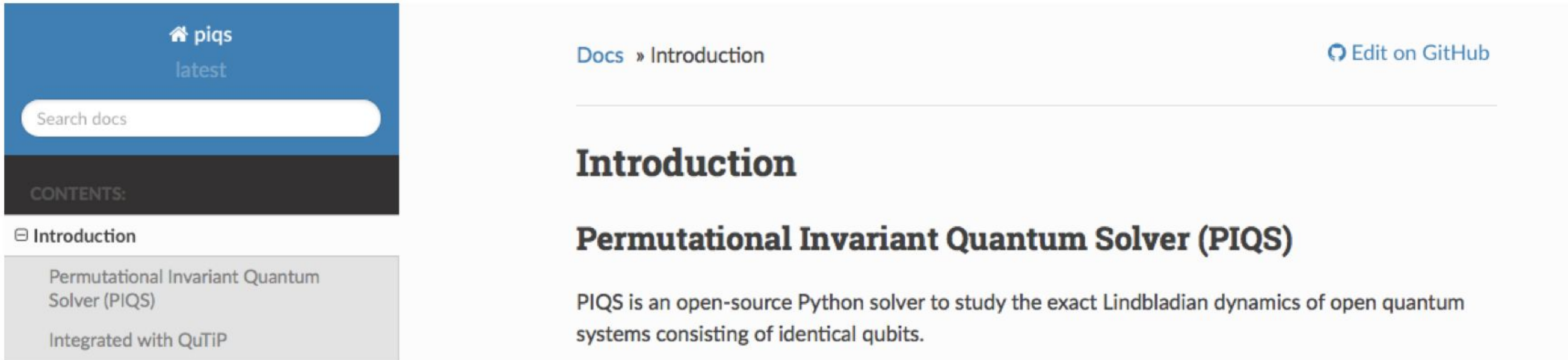
doc/source/conf.py

```
# -- Project information

project = 'piqs'

# The short X.Y version
version = ''
# The full version
release = '1.0'
```

Generate documentation

```
>> make html
```

Read the docs - host online



X

Unitary Fund

Distributing package: pip, conda

PyPI

conda-forge

CONDA-FORGE

Add setup information

piqs/setup.py

```
from setuptools import setup
setup(name=‘piqs',
      version='0.1',
      description=‘Permutational Invariant
              …)
```

meta.yml (recipe)

```
{% set name = "piqs" %}
{% set version = "1.0" %}

package:
  name: {{ name|lower }}
```

Make a package and üpload

```
>> python setup.py register sdist upload
```

Install from pip

```
>> pip install piqs
```

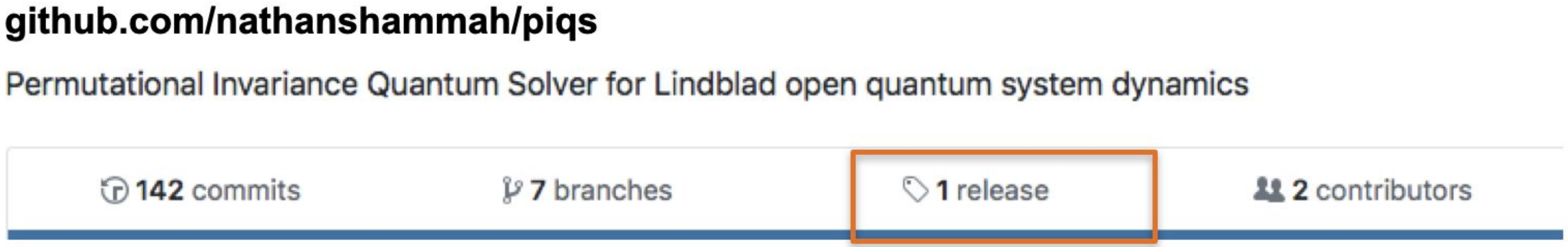
```
>> conda build
```

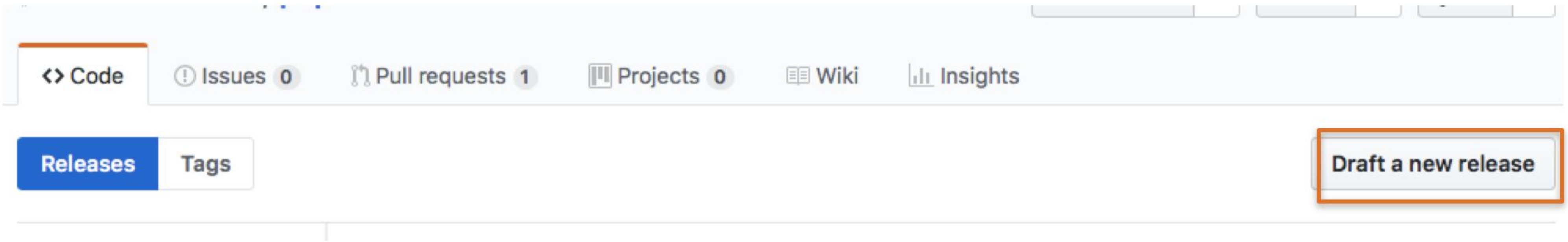
Add to a personal channel or conda-forge

```
>> conda install -c www.github.com/paul
```

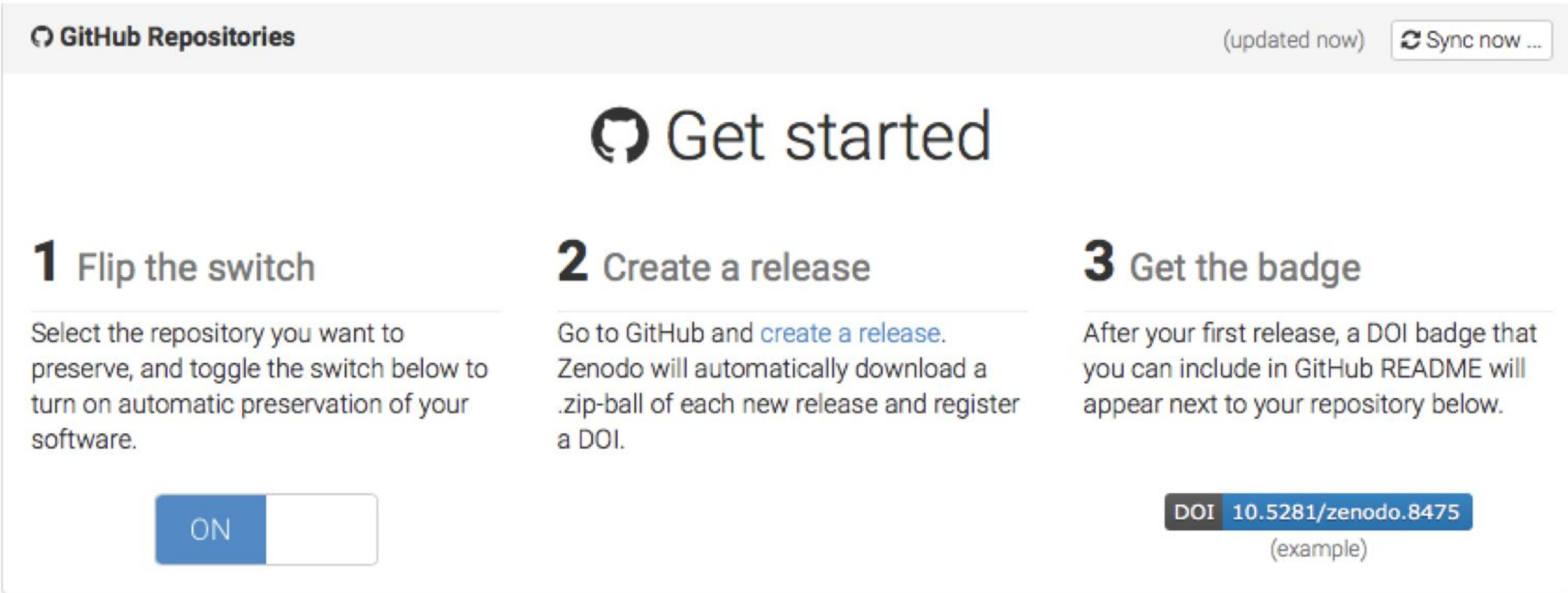
X

Publishing online: Release





Publishing online: Zenodo



Thank you!

discord.unitary.fund

nathan@unitary.fund