

---

# Diverse Community Around the Open Tools that Drive Open Science



Leah Wasser, Executive Director & Founder



ALFRED P. SLOAN  
FOUNDATION

---

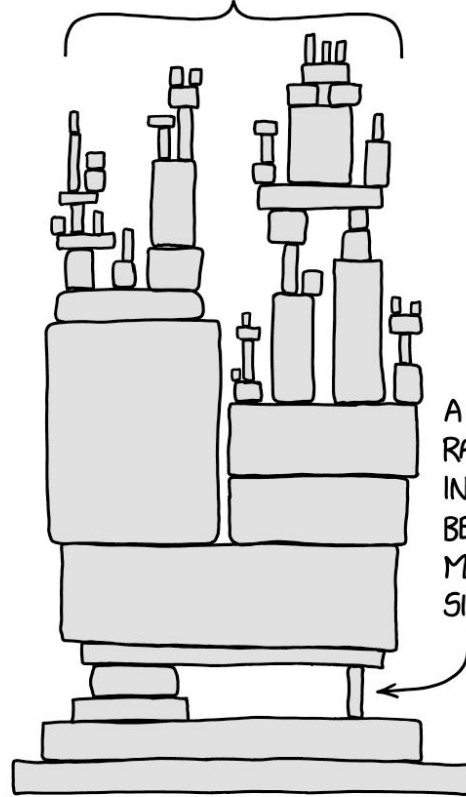
# About Me

- Phd Remote Sensing Ecology
- 3rd program that i've built

# Outline

- Open science & open source
- About pyOpenSci
  - Community driven packaging guide
  - Open peer review

ALL MODERN DIGITAL  
INFRASTRUCTURE



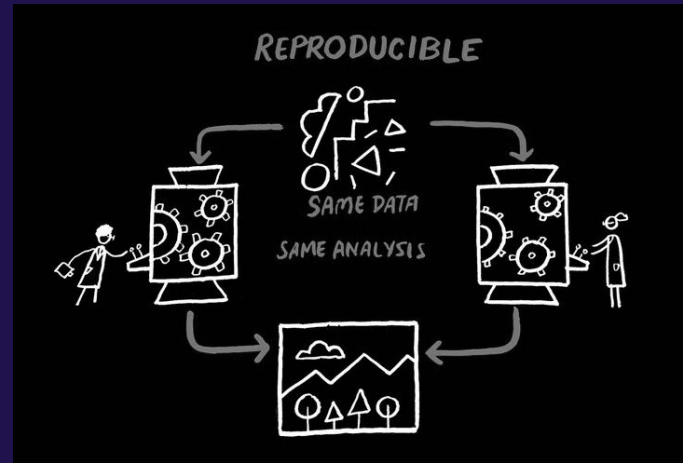
A PROJECT SOME  
RANDOM PERSON  
IN NEBRASKA HAS  
BEEN THANKLESSLY  
MAINTAINING  
SINCE 2003

---

**Billions to  
Trillions**  
Each year.

---

# Open Reproducible Science



- **Efficiency:** building on each other's work
- **Transparency** of methods

# Open Source Software Drives Open Science

When it works,  
well-documented &  
maintained.

- Allows scientists to create truly reproducible workflows
- **Diversity / level playing field:** not everyone can afford licenses

# Open source drives open science

- Tremendous effort goes into the development of each tool





# Open source is not just about software

- Community that values free and open software

---

# Issues that we face in science/research

## Inconsistent package quality

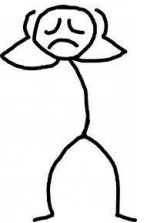
- Inconsistent maintenance of packages over time
- Inconsistent quality of documentation
- Not beginner friendly

## Community Misunderstanding

- Lack of understanding that tools they are using are created by **volunteer maintainers who believe in open source.**

## Truth

- Creating a healthy Python package is hard



---

# pyOpenSci builds diverse community around scientific software

*peer review, training & mentorship*





**2018**

Founded

**2019**

Started peer  
review

**2021**

Sloan Foundation  
Funding

**Fall 2022**

Launched as  
independent  
organization



# pyOpenSci Leadership

## Executive council



**Tracy Teal**  
Board chair



**Karen Cranston**

## DEIA & education council



**Ariane Sasso**

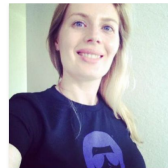


**Agustina Pesce**

## Advisory council



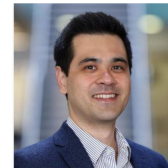
**Chris Holdgraf**  
UC Berkeley



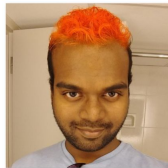
**Inessa Pawson**  
Numpy, OpenTeams Incubator



**Pradyun Gedam**  
PyPA, PSF, Bloomberg Python Infrastructure



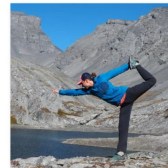
**Leonardo Uieda**  
University of Liverpool



**Yuvi Panda**



**Filipe Fernandes**



**Lindsey Heagy**  
@simpeg, @geoscixyz, @ubcgif, @z12c\_org

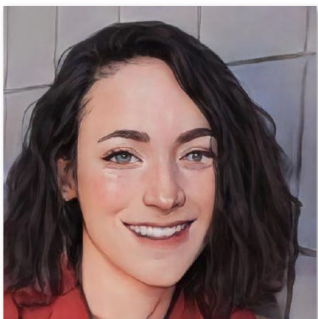
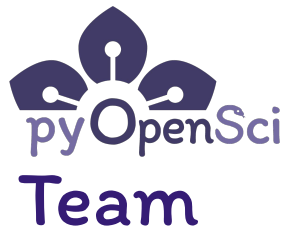


**Martin Fleischmann**



**Ivan Ogasawara**  
xmnlab

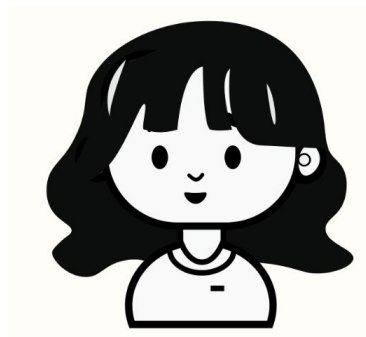
<https://www.pyopensci.org/our-community/>



**Leah Wasser**  
Executive Director &  
Founder



**Juanita Gomez**  
Community Content  
Coordinator



**New Hire**  
Community  
Manager (August  
2023 start)



# Community-driven Packaging Standards

<https://www.pyopensci.org/python-package-guide/>

---

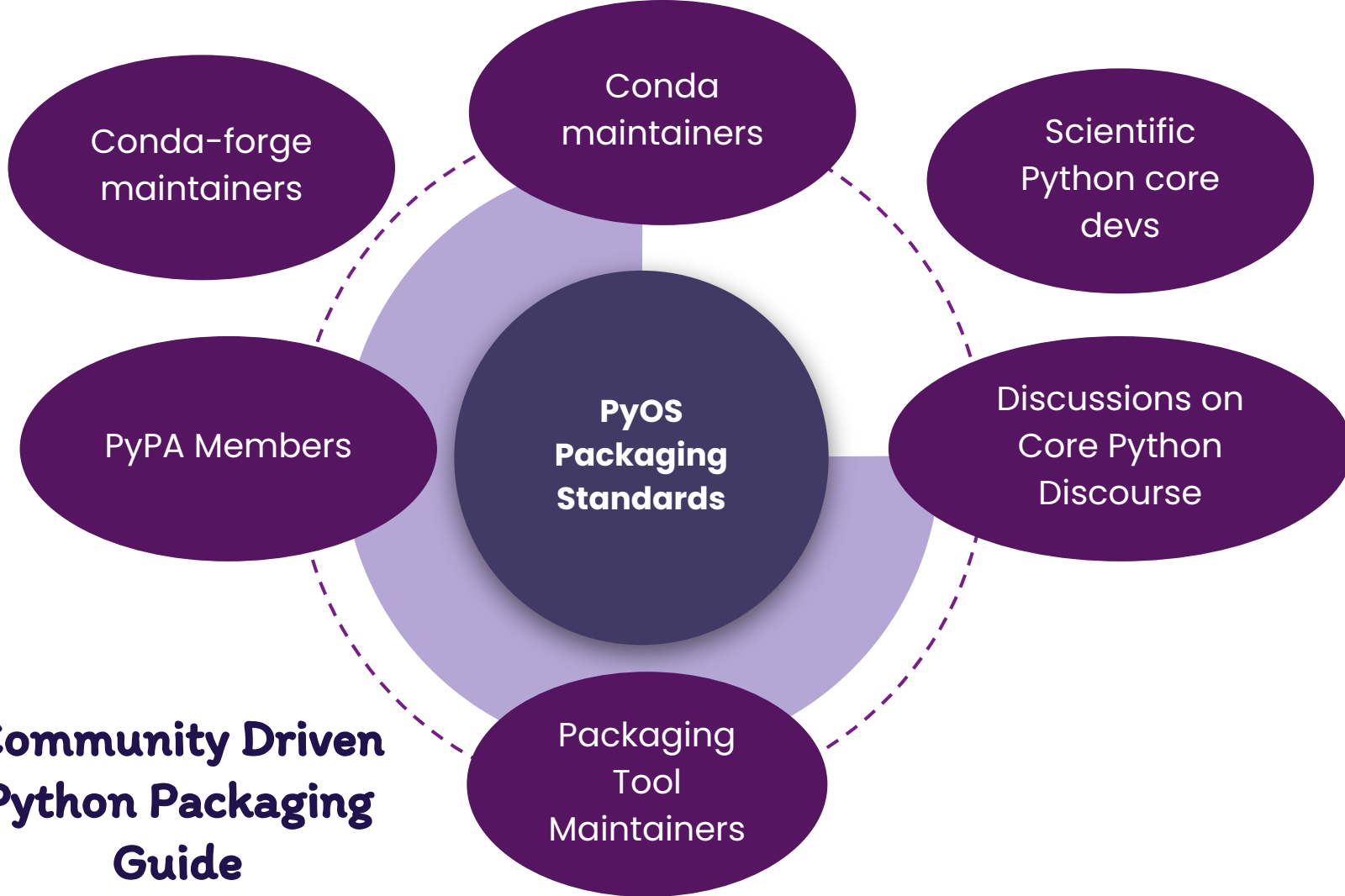


What packaging tool  
should I use???

How do I create a  
Python package?



# Community Driven Python Packaging Guide



# Python Packaging Guide

A community driven process

Talk with core experts

- Write a section of the guide

Semi-closed review

- \* Core experts review

Open Review 1

- Ping tool developers and maintainers
- Welcome broad community feedback

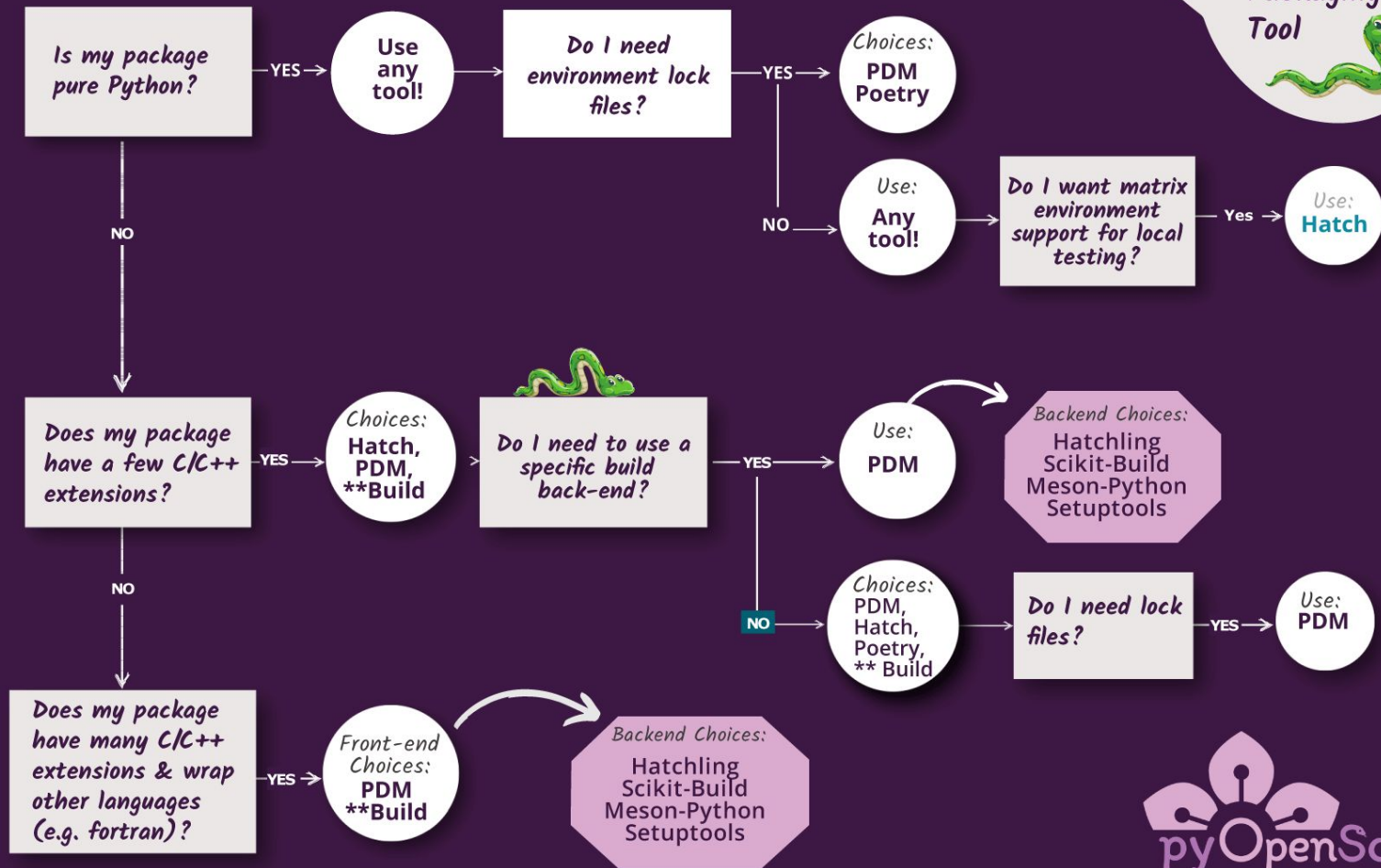
Open Review 2

- Welcome broad community feedback

Publish  
*(living document)*

[www.pyopensci.org/python-package-guide/](http://www.pyopensci.org/python-package-guide/)

Pure Python packages can use any back-end.  
Consider starting with a default back-end for the tool that you select (ie use pdm-back-end with the PDM front-end)



# Package tool features table

Feature	Flit	Hatch	PDM	Poetry
Default Build Back-end	Flit-core	hatchling	PDM	Poetry-core
Use Other Build Backends	✗	✗	✓	✗
Dependency management	✗	✗	✓	✓
Publish to PyPI	✓	✓	✓	✓
Version Control based versioning (using <code>git tags</code> )	✗	✓	✓	✓
Version bumping	✗	✓	✓	✓
Environment Management	✗	✓	✓	✓
More than one maintainer? (bus factor)	✗	✗	✗	✓

---

<https://www.pyopensci.org/python-package-guide>

<https://www.pyopensci.org/python-package-guide/>

---



# Software Peer Review

---



# Structure of a review (EIC checks) editor, 2 reviewers)

*Editor in Chief Performs Checks*



Submit  
PreSubmission  
Inquiry on  
[GitHub](#)

Submit Tool For  
Review  
[GitHub](#)

Package Review

pyOS standards

Conversation

 Package  
Accepted 

*Optional*

Fast Track in JOSS  
(if in scope)



✨ Package is a part of the pyOS ecosystem ✨

- Maintenance check-ins
- Community visibility
- Maintainer still “owns” tool

*Review mentorship available!*



# Editor In Chief

Performs initial basic  
checks

<https://www.pyopensci.org/software-peer-review/>

- [ ] **\*\*Installation\*\*** The package can be installed from
  - [ ] The package imports properly into a standard Python environment
  - [ ] **\*\*Fit\*\*** The package meets criteria for [fit](https://www.pyopensci.org/software-peer-review/fit)
  - [ ] **\*\*Documentation\*\*** The package has sufficient online documentation
    - [ ] User-facing documentation that overviews how to use the package
    - [ ] Short tutorials that help a user understand how to use the package
    - [ ] API documentation (documentation for your code's public interface)
  - [ ] Core GitHub repository Files
    - [ ] **\*\*README\*\*** The package has a `README.md` file with instructions on how to use the package
    - [ ] **\*\*Contributing File\*\*** The package has a `CONTRIBUTING.md` file with instructions on how to contribute to the package
    - [ ] **\*\*Code of Conduct\*\*** The package has a `Code of Conduct` file with instructions on how to interact with the community
    - [ ] **\*\*License\*\*** The package has an [OSI approved license](https://opensource.org/licenses/OSI-Approved)
- NOTE: We prefer that you have development instructions in the README file
- [ ] **\*\*Issue Submission Documentation\*\*** All of the information needed to submit an issue is in the README file
  - [ ] **\*\*Automated tests\*\*** Package has a testing suite and CI/CD pipeline

# Structure of a review (EIC checks) editor, 2 reviewers)

*Editor in Chief Performs Checks*



Submit  
PreSubmission  
Inquiry on  
[GitHub](#)

Submit Tool For  
Review  
[GitHub](#)

Package Review

pyOS standards

Conversation

 Package  
Accepted 

*Optional*

Fast Track in JOSS  
(if in scope)



✨ Package is a part of the pyOS ecosystem ✨

- Maintenance check-ins
- Community visibility
- Maintainer still “owns” tool

*Review mentorship available!*

# PyGMT: A Python interface for the Generic Mapping Tools #43

Open

3 of 9 tasks

weiji14 opened this issue on Jul 22, 2021 · 59 comments



weiji14 commented on Jul 22, 2021 · edited by lwasser ▾



Submitting Author: Wei Ji Leong (@weiji14)

Package Name: PyGMT

One-Line Description of Package: A Python interface for the Generic Mapping Tools

Repository Link (if existing): <https://github.com/GenericMappingTools/pygmt>

Version submitted:

Editor: @lwasser

Reviewer 1: @jbusecke

Reviewer 2: @SimonMolinsky

Archive: [Zenodo Archive](#)

Version accepted: V 0.7.0

Date accepted (month/day/year): 9/1/2022

# pyOpenSci Editorial Team



**Filipe**

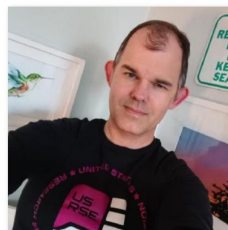
Editor



**Ariane Sasso**

Editor

@hpi-dhc



**David Nicholson**

Editor in Chief



**Chiara Marmo**

Editor



**C. Titus Brown**

Editor

University of California,  
Davis



**Meenal Jhajharia**

Guest Editor

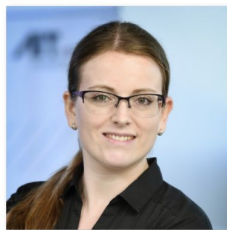
Uvornoting



**Jenny Palomino**

Editor

Google



**Anita Graser**

Editor



**Alex Batisse**

Editor



**Tom Russell**

Editor

University of Oxford



**Szymon Moliński**

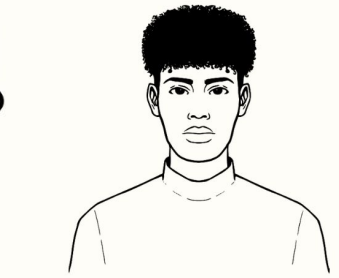
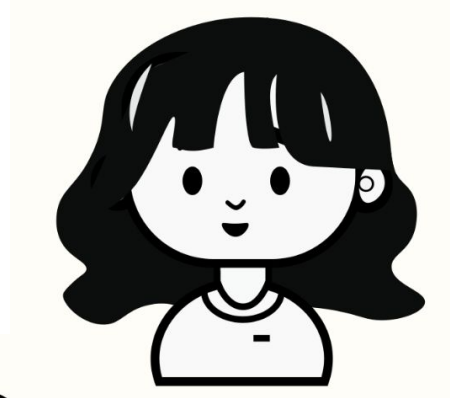
Editor

Sare



---

# Diversity: Reviewer teams



---

# Peer Review Mentorship





Robaina commented yesterday

Author



Hi @NickleDave,

thank you for your careful first review of the project, here are my thoughts:

- We would want the docs to be a Sphinx site, not a wiki, as stated in our editor checks [here](#) although overall the docs are very readable. I would also recommend copying example snippets of using the cli so they are one of the very first things a potential user sees on your README; they should know immediately what the interface actually looks like

That makes sense. Will port the wiki to a Sphinx site. I agree on including cli snippets!

- We would definitely want you to use a pyproject.toml that declares your dependencies as metadata, instead of the deprecated setup.py approach. Ideally with new-style metadata as described in the `flit` docs: [https://flit.pypa.io/en/stable/pyproject\\_toml.html#new-style-metadata](https://flit.pypa.io/en/stable/pyproject_toml.html#new-style-metadata). For more on why we require a pyproject.toml, see [this post from Titus Brown](#), and [this much more detailed post from Paul Anzel](#).

Will do. A good opportunity to finally get rid of the deprecated setup.py.

- Part of the purpose of pyOpenSci is to standardize project structure; there's no right answer but it is definitely uncommon to see a directory with the package name that then has a `src/` directory inside of it. Usually you would either just have your source code in `pynteny` or you would have `src/pynteny`. See this post: <https://hynek.me/articles/testing-packaging/>. Similarly, while people do sometimes have tests along code, it's often preferable to completely isolate your tests so you know they are testing *installed* code. I would suggest switching to `src/pynteny` or `pynteny` and moving your `tests` to the project root.

# pynten

Presubmission

- Move metadata setup.py → pyproject.toml
- Moved docs from wiki → sphinx
- Helped with install issues



# Domain-specific partnerships

<https://www.pyopensci.org/software-peer-review/partners/pangeo.html>

maintainers  
Submit Tool For  
Review

(2 reviewers)  
Package Review

**pyOS standards**

**Pangeo standards**

 **Package  
Accepted** 

JOSS: Fast Track



✨ **Package is a part of the pyOS ecosystem** ✨

- Annual maintenance check-ins,
- Community support
- Community visibility as a vetted tool
- Catalog of vetted tools

✨ **Package is a vetted Pangeo affiliated tool** ✨



---

# Domain specific partnerships

## ***pyOpenSci provides:***

- Structure & processes for peer review
  - Cross ecosystem enforcement of packaging standards
  - Visibility for your packages
  - Cross ecosystem tool redundancy checks
-

# Cross-domain promotion of packages

## Our accepted Python open source packages

The packages below have already been through our open peer review process and are accepted as pyOpenSci packages.

Search

show all data munging data extraction data retrieval data-visualization education geospatial reproducibility

### xclim

Zeitsperre, tlogan2000, aulemahal

Climate indices computation package based on Xarray

[View Code](#)  
[View Docs](#)

### crowsetta

NickleDave

A Python tool to work with any format for annotating animal vocalizations and bioacoustics data.

[View Code](#)  
[View Docs](#)

### Pynteny

Robaina

Query sequence database by HMMs arranged in predefined synteny structure

[View Code](#)  
[View Docs](#)

### Devicely

Ariane Sasso

A Python package for reading, timeshifting and writing sensor data

[View Code](#)  
[View Docs](#)

### PyGMT

Wei Ji Leong

A Python interface for the Generic Mapping Tools

[View Code](#)  
[View Docs](#)

### Jointly

Arne Herdick

Jointly is a python package for synchronizing sensors with accelerometer data

[View Code](#)  
[View Docs](#)



**Community  
Partnerships**



Scientific Python

Thank you!



 @leahawasser@fosstodon.org

 leahawasser

leah@pyopensci.org

 @pyopensci@fosstodon.org

 pyopensci

## Ways to get involved:

- [Sign up for discourse to get latest updates!](#)
- [Sign up to be a reviewer](#)
- [Submit a package for review](#)
- Review our peer review guide
- Review our packaging guide
- Post a packaging question on discourse
- Spread the word!