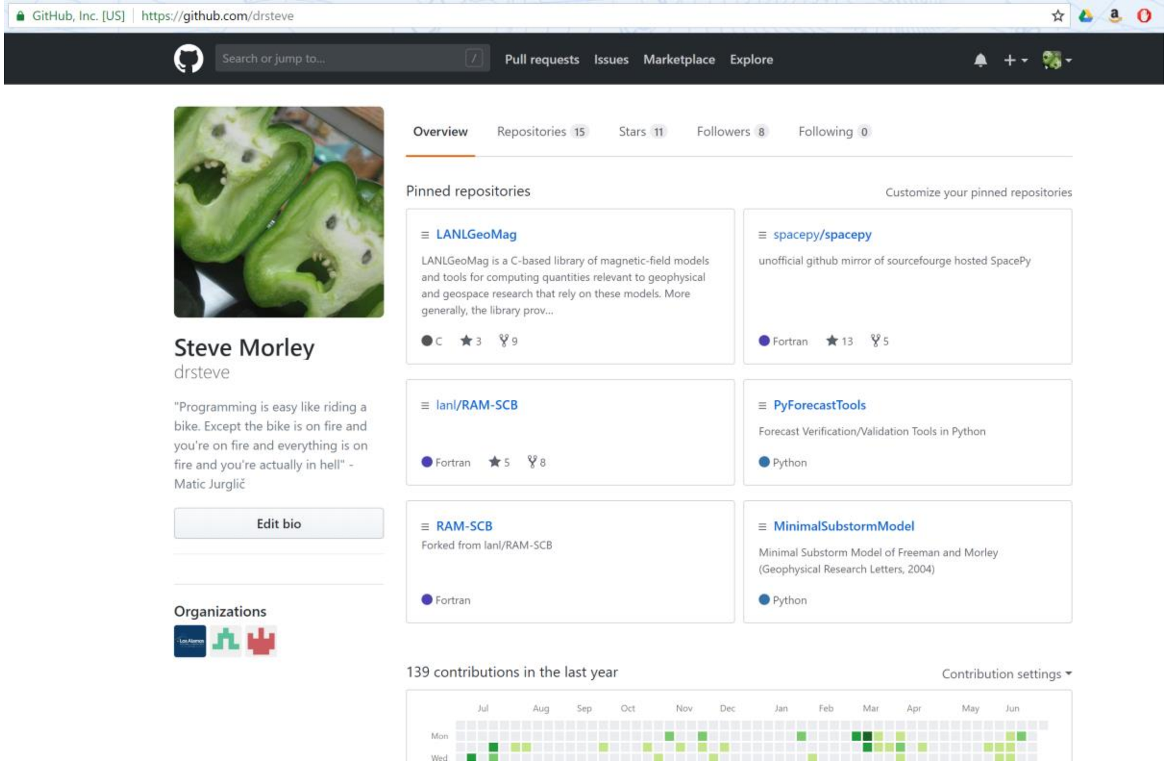
Code sharing, Discovery and Credit

Steven K. Morley

UNCLASSIFIED

Shameless self-promotion

github.com/drsteve – please check out spacepy and others…



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

Search or jump to... Pull requests Issues Marketplace Explore

Overview Repositories 15 Stars 11 Followers 8 Following 0

Steve Morley

drsteve

"Programming is easy like riding a bike. Except the bike is on fire and you're on fire and everything is on fire and you're actually in hell" - Matic Jurglič

Edit bio

Organizations

Pinned repositories

Customize your pinned repositories

LANLGeoMag

LANLGeoMag is a C-based library of magnetic-field models and tools for computing quantities relevant to geophysical and geospace research that rely on these models. More generally, the library prov...

● C ★ 3 9

spacepy/spacepy

unofficial github mirror of sourcefourge hosted SpacePy

● Fortran ★ 13 5

lanl/RAM-SCB

● Fortran ★ 5 8

PyForecastTools

Forecast Verification/Validation Tools in Python

● Python

RAM-SCB

Forked from lanl/RAM-SCB

● Fortran

MinimalSubstormModel

Minimal Substorm Model of Freeman and Morley (Geophysical Research Letters, 2004)

● Python

139 contributions in the last year

Contribution settings

SpacePy

- Python/Fortran/C

PyForecastTools

- Python

RAM-SCB

- Fortran

LANLGeoMag

- C

MinimalSubstormModel

- Python

I made this

Officially out in the open

- Choose a license*
  - see https://opensource.org/licenses
  - Most of my projects use BSD license
  - GPL can restrict use…
- Choose a host
  - E.g. GitHub, SourceForge, BitBucket
- Make it discoverable!
  - LANLGeoMag is AUL 9, in use for several operational needs, but is not widely known

UNCLASSIFIED

I made this

Officially out in the open

2009:

- SpacePy is born
- VCS using local SVN

2010:

- Open-source approval from LANL (PSF license)
- Code migrated to SourceForge
- SpacePy presentation at SciPy conference!

2011:

- Migrate to git
- SciPy conference proceedings published

Hey, you! I made this!

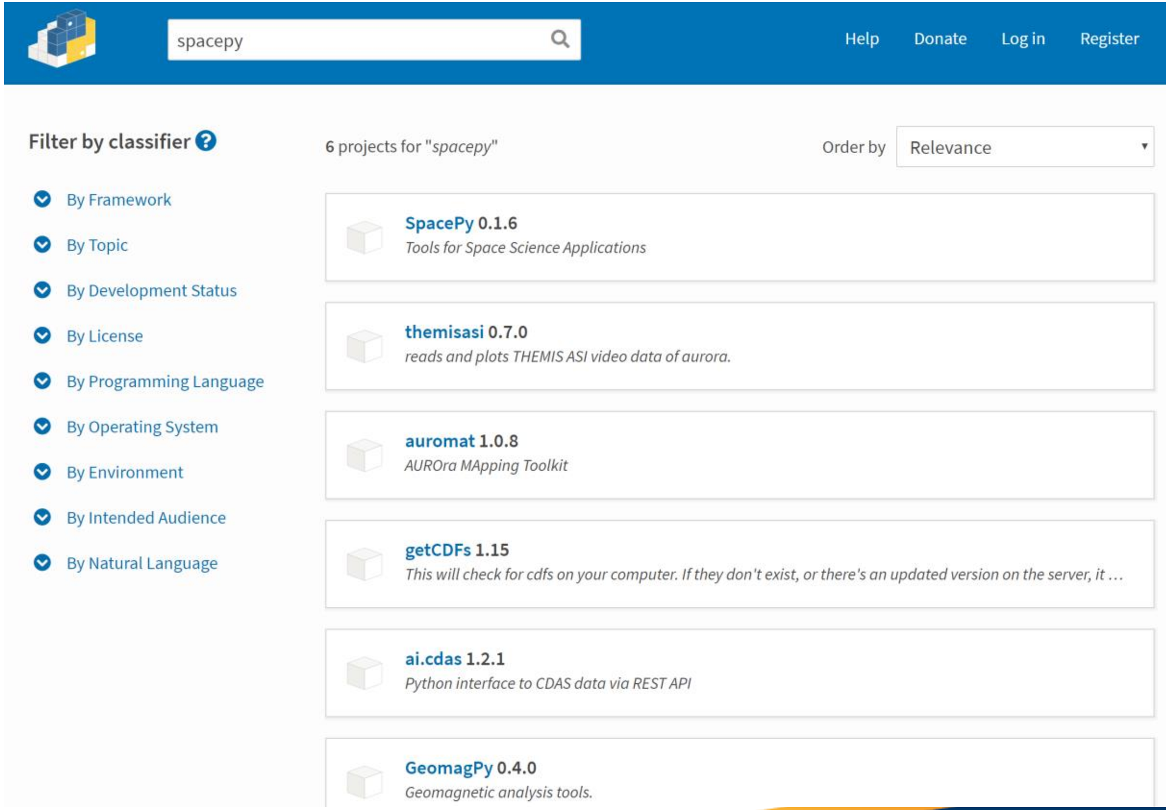
Making code discoverable

For Python, PyPI is key!

- Discoverability
- Installability

Strong web presence with good keywords helps discoverability

Often "discoverability" is just "what comes up when I GDS?"



6 projects for "*spacepy*"

SpacePy 0.1.6
Tools for Space Science Applications

themisasi 0.7.0
reads and plots THEMIS ASI video data of aurora.

auromat 1.0.8
AUROra MApping Toolkit

getCDFs 1.15
This will check for cdfs on your computer. If they don't exist, or there's an updated version on the server, it ...

ai.cdas 1.2.1
Python interface to CDAS data via REST API

GeomagPy 0.4.0
Geomagnetic analysis tools.

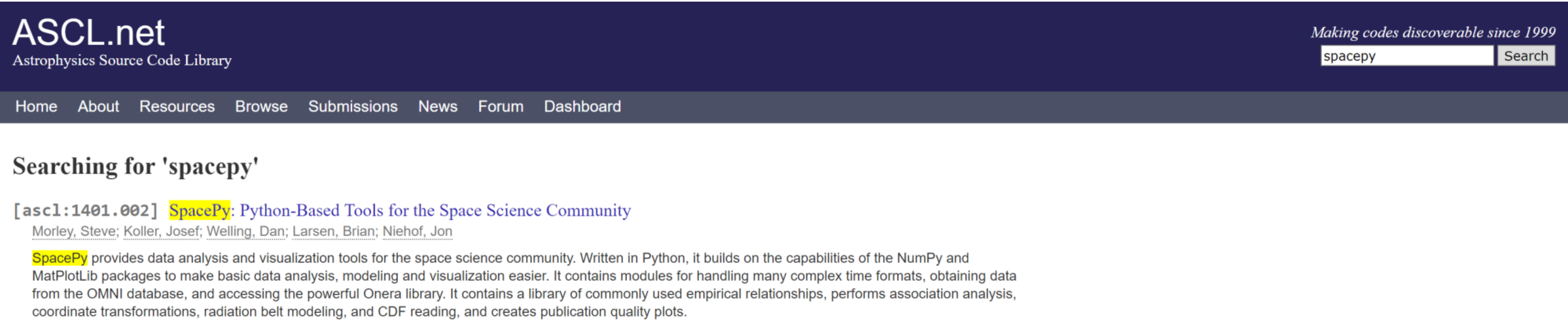
UNCLASSIFIED

Hey, you! I made this!

Making code discoverable

Other options:

- *Astrophysics Source Code Library*
- *code.nasa.gov (for NASA employees)*
- *Journal articles*



Searching for 'spacepy'

[ascl:1401.002] SpacePy: Python-Based Tools for the Space Science Community

Morley, Steve; Koller, Josef; Welling, Dan; Larsen, Brian; Niehof, Jon

SpacePy provides data analysis and visualization tools for the space science community. Written in Python, it builds on the capabilities of the NumPy and MatPlotLib packages to make basic data analysis, modeling and visualization easier. It contains modules for handling many complex time formats, obtaining data from the OMNI database, and accessing the powerful Onera library. It contains a library of commonly used empirical relationships, performs association analysis, coordinate transformations, radiation belt modeling, and CDF reading, and creates publication quality plots.

where credit is due…

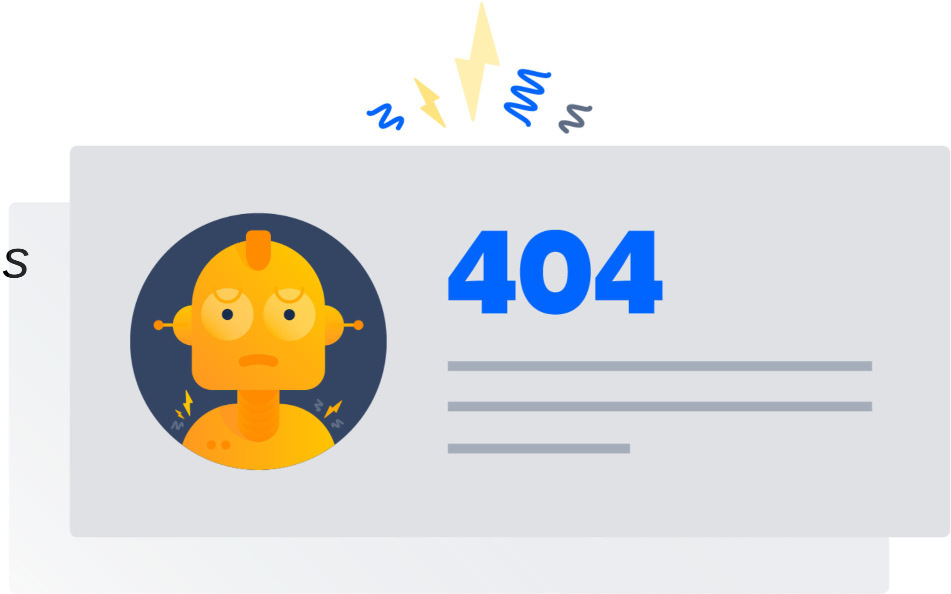
Making code citeable

Digital Object Identifiers (DOIs) were introduced to provide permanent links to online resources

- *URLS can change…*

But my code is on GitHub!

- *GitHub provides a URL*
- *ASCL records URLs*
- *Journal articles have DOIs, but aren't a good way of storing code*

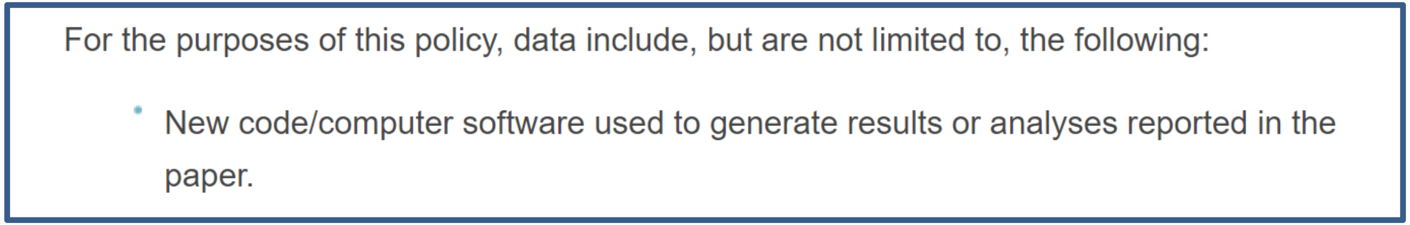


UNCLASSIFIED

Your code is data

Various variants of versioning

AGU requires that data are made available at the time of publication



For the purposes of this policy, data include, but are not limited to, the following:

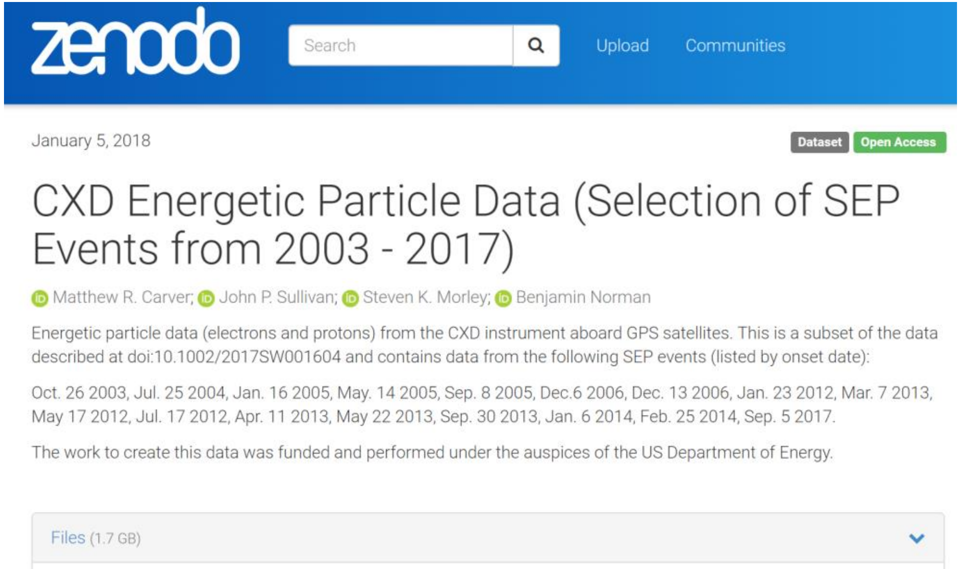
- New code/computer software used to generate results or analyses reported in the paper.

Open data repositories that issue DOIs include:

- Zenodo (zenodo.org)
- Dryad (datadryad.org)
- Some institutions and gov't agencies

Your code is data

Examples of data and code at Zenodo



January 5, 2018

Dataset Open Access

CXD Energetic Particle Data (Selection of SEP Events from 2003 - 2017)

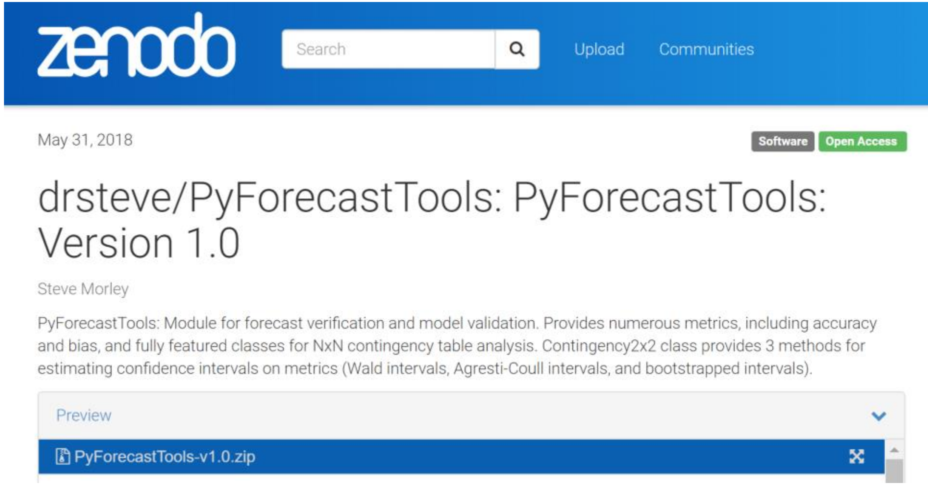
Matthew R. Carver; John P. Sullivan; Steven K. Morley; Benjamin Norman

Energetic particle data (electrons and protons) from the CXD instrument aboard GPS satellites. This is a subset of the data described at doi:10.1002/2017SW001604 and contains data from the following SEP events (listed by onset date):

Oct. 26 2003, Jul. 25 2004, Jan. 16 2005, May. 14 2005, Sep. 8 2005, Dec.6 2006, Dec. 13 2006, Jan. 23 2012, Mar. 7 2013, May 17 2012, Jul. 17 2012, Apr. 11 2013, May 22 2013, Sep. 30 2013, Jan. 6 2014, Feb. 25 2014, Sep. 5 2017.

The work to create this data was funded and performed under the auspices of the US Department of Energy.

Files (1.7 GB)



May 31, 2018

Software Open Access

drsteve/PyForecastTools: PyForecastTools: Version 1.0

Steve Morley

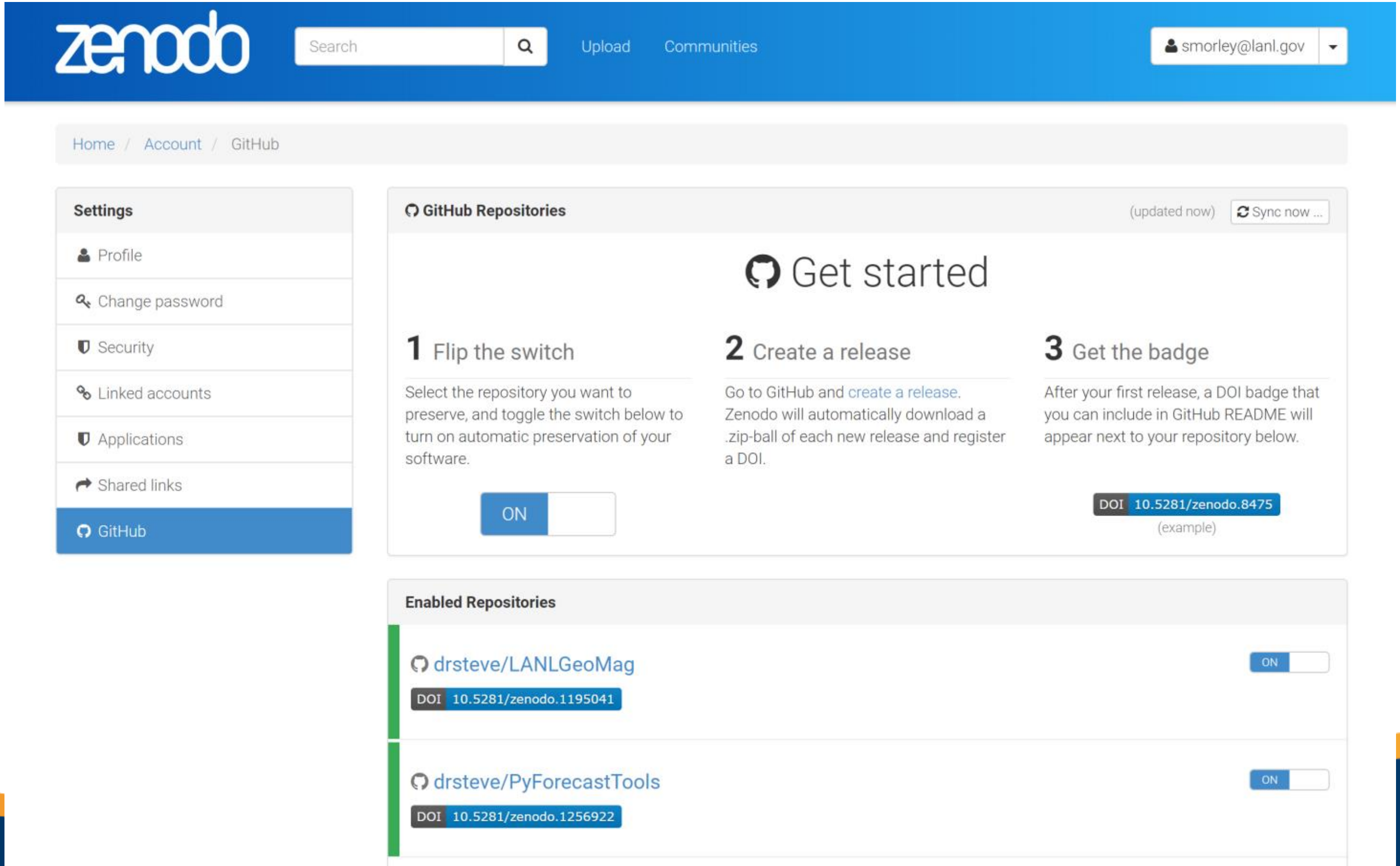
PyForecastTools: Module for forecast verification and model validation. Provides numerous metrics, including accuracy and bias, and fully featured classes for NxN contingency table analysis. Contingency2x2 class provides 3 methods for estimating confidence intervals on metrics (Wald intervals, Agresti-Coull intervals, and bootstrapped intervals).

Preview

PyForecastTools-v1.0.zip

Zenodo-GitHub integration

When should a “release” be made?



GitHub Repositories

(updated now) Sync now ...

Get started

1 Flip the switch

Select the repository you want to preserve, and toggle the switch below to turn on automatic preservation of your software.

ON

2 Create a release

Go to GitHub and create a release. Zenodo will automatically download a .zip-ball of each new release and register a DOI.

3 Get the badge

After your first release, a DOI badge that you can include in GitHub README will appear next to your repository below.

DOI 10.5281/zenodo.8475

(example)

Enabled Repositories

drsteve/LANLGeoMag ON

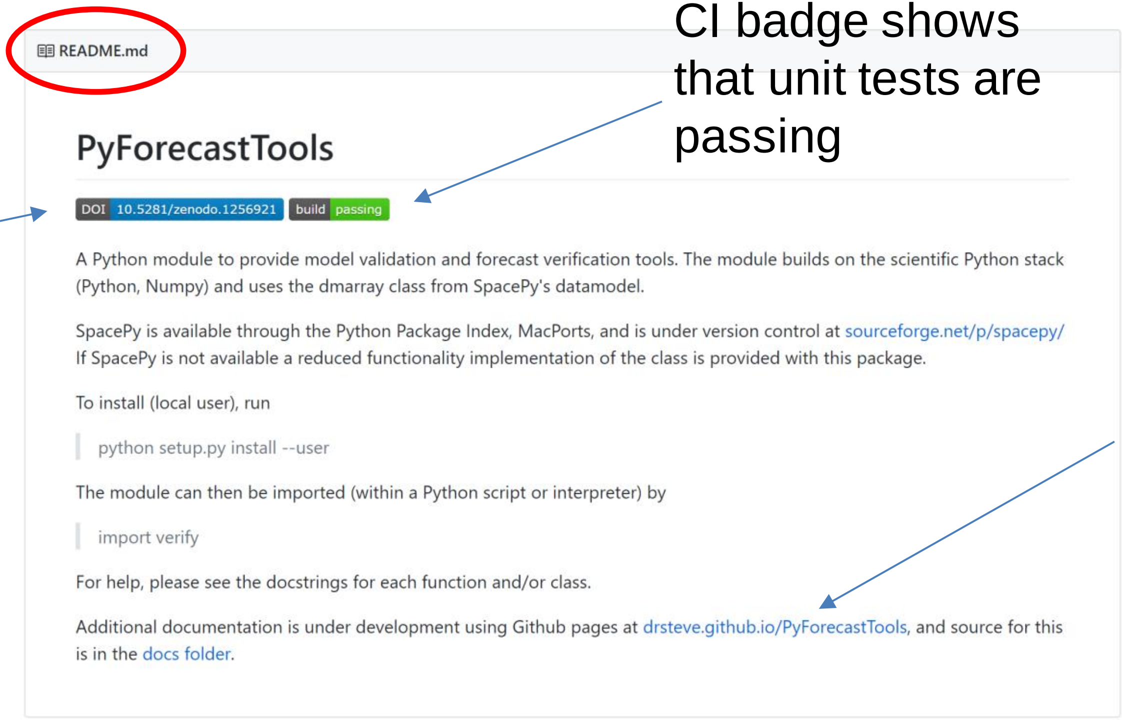
DOI 10.5281/zenodo.1195041

drsteve/PyForecastTools ON

DOI 10.5281/zenodo.1256922

I found your GitHub

Now what?



README.md

PyForecastTools

DOI 10.5281/zenodo.1256921 build passing

A Python module to provide model validation and forecast verification tools. The module builds on the scientific Python stack (Python, Numpy) and uses the dmarray class from SpacePy's datamodel.

SpacePy is available through the Python Package Index, MacPorts, and is under version control at sourceforge.net/p/spacepy/ If SpacePy is not available a reduced functionality implementation of the class is provided with this package.

To install (local user), run

> python setup.py install --user

The module can then be imported (within a Python script or interpreter) by

> import verify

For help, please see the docstrings for each function and/or class.

Additional documentation is under development using Github pages at drsteve.github.io/PyForecastTools, and source for this is in the docs folder.

CI badge shows that unit tests are passing

DOI badge links to Zenodo

Where are the docs?

Please publish me!

Where can code papers go?

- JGR: Space Physics
  - E.g., Stoneback et al. (2018), PYSAT: Python Satellite Data Analysis Toolkit
- SciPy Conference Proceedings
  - E.g., Morley et al. (2010), SpacePy: A Python-based library of tools for the space sciences
- Journal of Open Source Software
  - “Developer-friendly”: Peer-reviewed short summaries with code
  - Reviews are done as GitHub tickets; code is archived on Zenodo
- Journal of Open Research Software
- Journal of Computational Science
- Geoscientific Model Development

The citation stealer

Pitfalls of proliferating products

So my code is open source

- It’s under version control at github
- I have it indexed at PyPI and ASCL
- My paper has been published and archived at Zenodo

So now what do I cite?

- GitHub URL? (Quasi-citeable)
- The Zenodo archive? (Citeable; Has a DOI)
- The ASCL record? (Citeable; Does not have a DOI)
- The journal article? (Citeable; Has a DOI)

Now imagine there have been several “releases”, each with multiple DOIs

The citation stealer

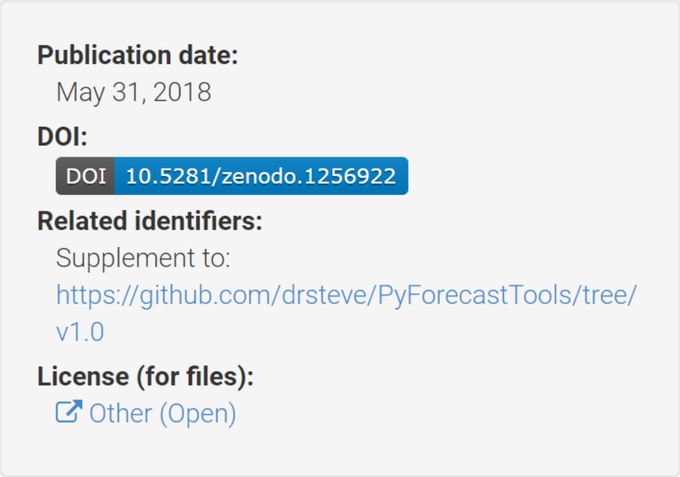
Pitfalls of proliferating products

Publicize preferred citation.

Make preferred citation data available in code!

```
>>> import spacepy
>>> spacepy.__citation__
```

Zenodo provides a “master” DOI that always points to the latest version.

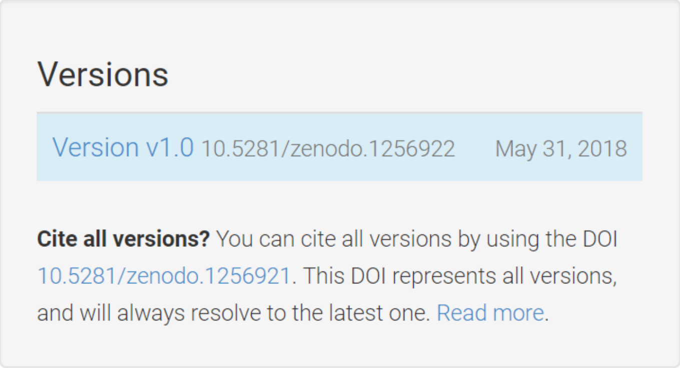


Publication date:
May 31, 2018

DOI:
DOI 10.5281/zenodo.1256922

Related identifiers:
Supplement to:
https://github.com/drsteve/PyForecastTools/tree/v1.0

License (for files):
Other (Open)



Versions

Version v1.0 10.5281/zenodo.1256922 May 31, 2018

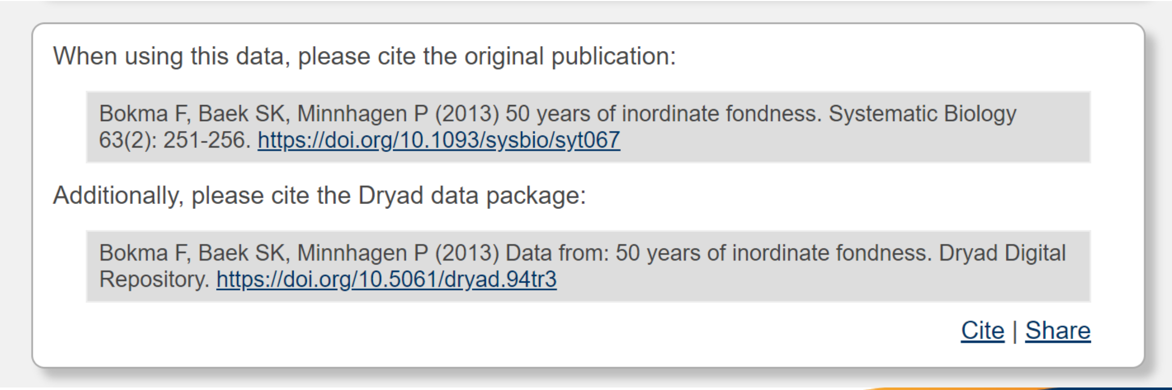
Cite all versions? You can cite all versions by using the DOI 10.5281/zenodo.1256921. This DOI represents all versions, and will always resolve to the latest one. Read more.

The citation stealer

Pitfalls of proliferating products

Dryad includes preferred citation data

Can be done in code, e.g., in README markdown file. Good for GitHub projects.



When using this data, please cite the original publication:

Bokma F, Baek SK, Minnhagen P (2013) 50 years of inordinate fondness. Systematic Biology 63(2): 251-256. https://doi.org/10.1093/sysbio/syt067

Additionally, please cite the Dryad data package:

Bokma F, Baek SK, Minnhagen P (2013) Data from: 50 years of inordinate fondness. Dryad Digital Repository. https://doi.org/10.5061/dryad.94tr3

Code Sharing

Releasing code is not enough

Logos and merch help too!

- Release code with open-source license
- Make code discoverable
- Make code citeable
- Engage with, and support, your community

- Establish community best practices to make all this easier!

Make code useable and trustable!
(Document and unit-test!)