sunpy
A Community Python Library for Solar Physics

1 Introduction and motivation for SunPy

IDL SolarSoft is the current paradigm for scientific analysis in solar physics. IDL is a closed-source, proprietary software and SolarSoft consists of a list of routines primarily written in IDL. The vast majority of researchers still use IDL because legacy solar physics code from the 90's was written in IDL. However, there are significant drawbacks to this paradigm. SolarSoft has very little documentation, none of which is in a standardized format. There is also a lack of version control which makes it exceedingly difficult to reproduce your colleague's scientific results. Additionally, obtaining an IDL license typically costs $\sim$ \$1000. In some departments in India and Nepal, creating a solar physics program was simply not economically viable since the cost of IDL licenses were a significant fraction of a department's budget.

In response to these shortcomings, SunPy was created on 28th of March 2011 by a few scientists and developers and NASA Goddard. SunPy aims *"to facilitate and promote the use and development of a community-led, free and open-source solar data-analysis software based on the scientific Python environment."* It has since become a recognized community package with over a 100 Contributors and integrates well with other libraries in Python's scientific computing ecosystem."

2 The SunPy community

As of 20 Nov 2019, SunPy currently has 119 contributors, 19 of whom have more than 100 commits. Communication happens through a mailing list, Github issues and a lively chatroom where response times are short. The main SunPy developers also attend the Python in Heliophysics conferences; the most recent one happened a few weeks ago in Boulder. This is a 2-3 day conference in which developers from SunPy and a few other open source packages meet to discuss future avenues and decide on short and long-term goals.

Figure 1 shows SunPy's activity over the past month. Most of the commits are minor bug-fixes or edits in documentation. More substantial changes to SunPy happen through SunPy Enhancement Proposals (SEP). Anyone can submit a proposal upon which it gets reviewed and voted on by the SunPy board of directors. Since 2016, SunPy has also had

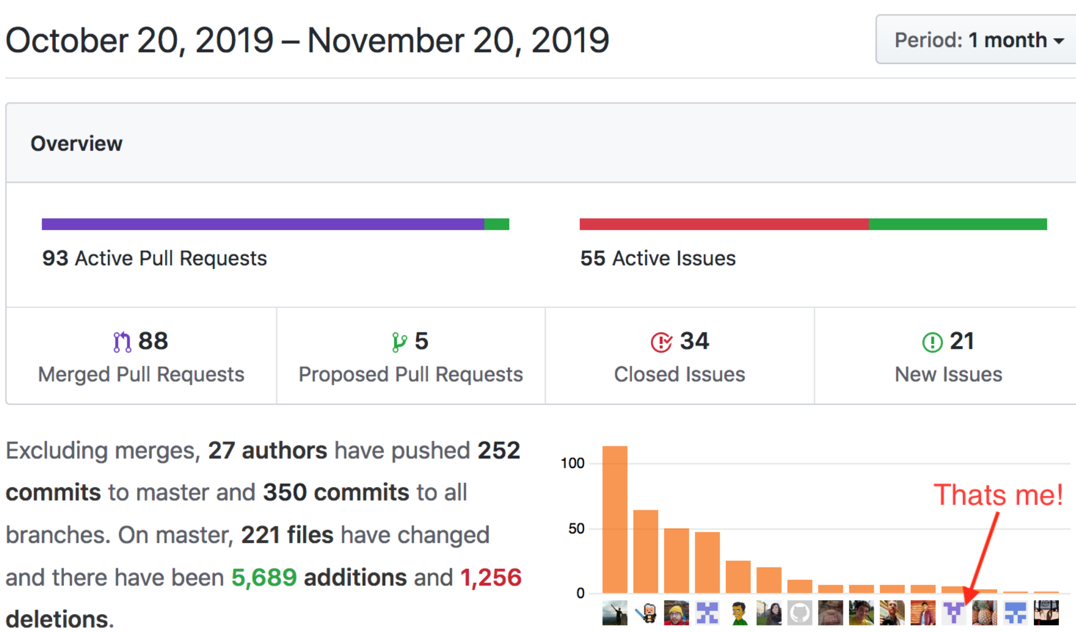

Figure 1: SunPy's activity over the past month

students routinely participate in Google summer of Code. These projects have usually involved a student contributing a significant new feature to SunPy.

Organizational structure and funding

The lead developer of SunPy is Stuart Mumford, who is currently at the University of Sheffield. Stuart has committed 15% of the total commits in SunPy. He is currently being paid by the National Solar Observatory (indirectly by the NSF) to write data reduction code and well as maintain and contribute to SunPy. The deputy lead developer is Nabil Freij who is currently a freelance Python developer but did research in Solar Physics for his PhD. There is also a board of directors who guide the overall vision of SunPy and work with the lead developer to implement it. SunPy is also a sponsored project of NumFOCUS.

3 Performance and capability needs of key stakeholders

SunPy's users are primarily researchers in solar physics. Their priorities are gathering and analyzing data and publishing results in a time-efficient manner. These goals align closely with SunPy's mission statement. However, there are still a few routines in IDL Solarsoft for which standardized implementations in SunPy don't exist. Hence, much of SunPy's future efforts involve writing the remaining routines.

SunPy's performance is also noticeably better than IDL since it uses NumPy's optimized code to deal with arrays. In a recent paper, the authors found that their methods using SunPy "produces results many times faster than typical DEM methods"

Researchers also don't migrate to SunPy due to the overhead cost of learning Python. However, short Python tutorials at solar physics conferences are becoming increasingly common. There are also "IDL to Python" Jupyter notebooks.

4 Planned contribution

To become more familiar with the Github workflow, I have submitted 2 pull requests to SunPy. My first PR was a trivial correction of a spelling mistake in a function. Next, I noticed that there was a documentation error in an example code. I created an issue and followed up with a PR to fix it. I'm currently working on a more significant contribution described below:

4.1 Implementing atmospheric refraction in SunPy

Figure 2 shows how the Sun can look dramatically different at different wavelengths. Aligning multi-wavelength images is a routine problem in processing data. Image alignment is further complicated by the fact that Earth's atmosphere refracts different wavelengths of light by different amounts; this shifts one image relative to another. My proposed contribution is to account for this effect. I plan to calculate the atmospheric offsets in a coordinate system appropriate for scientists who take observations at telescopes. I have posted this proposal on the developers mailing list and recently received support for this idea. I'm working on the code and hope to have a draft PR submitted soon.

Figure 2: The Sun looks very different at different wavelengths. The left image is taken at 1930 nanometers and right taken at 3040 nanometers.