The PyHC Open Science Experiment: A PyHC session led by Rebecca Ringuette

Rebecca Ringuette¹

¹Affiliation not available

August 4, 2023

The PyHC Open Science Experiment

Rebecca Ringuette and many others
NASA GSFC Center for HelioAnalytics

Presented at the 2023 PyHC Spring Meeting at LASP/CU Boulder in Boulder, CO.
Session Outline

• Invited presentation on the Open Science Framework (10-15 min)
  Presented by Gretchen Geugeun

• Project Introduction (15-20 min)
  • The PyHC Open Science Experiment
  • Project Tour

• Open Science and Heliophysics Infrastructure (5-10 min)

• Discussion (45 min): What PyHC software changes are needed to better support this project and, more generally, open science? What funding is needed to complete these tasks?
The Open Science Framework
Presented by Gretchen Geugeun
Link to slides: https://docs.google.com/presentation/d/1vtSmbsDweTLmS8wGgfwNu9aDQMZ3iMnPQeQTlKjje_M/edit?usp=sharing
Project Introduction: The PyHC Open Science Experiment

• The PyHC executable paper demonstrated:
  • How to **collaborate** between software developers/engineers and scientists,
  • How to **use multiple PyHC packages** to perform a science analysis,
  • How to produce an **executable paper** in Heliophysics, and
  • How such a collaboration **supports open science**.

• The goals for this work are to:
  • Apply the workflow developed to a **full-scale science** problem, specifically expanding the 2015 challenge with new data from MMS (https://ccmc.gsfc.nasa.gov/challenges/gem-magnetopause/),
  • Demonstrate how to **perform open science** in Heliophysics, and
  • Improve and **develop modern infrastructure** to streamline collaboration and contributions.
Project Introduction: The PyHC Open Science Experiment

**GEM Science Plan**

- Expand to include multiple time ranges of MMS data where magnetopause crossings occurred (retrieved with *pySPEDAS*),
- Generate the predictions using the empirical Shue model (using *SpacePy*),
- Generate flythrough results for each contributed physics-based model output stored in s3 (via *Kamodo*),
- Encourage the community to provide metrics calculation scripts using the flythrough results (built on *PlasmaPy*), and
- Provide a platform where all contributors can search and reuse all components (on *HelioCloud*).
- Multiple members of the community are expected to lead portions of the project and produce multiple papers, including a summary paper (coordinated on the *Open Science Framework*).
Project Introduction: The PyHC Open Science Experiment

Open Science goal

• Perform the work **in the open** from the beginning,
• **Demonstrate** how to perform open science to the Heliophysics community and various agencies and nations,
• **Develop** any lacking infrastructure along the way (as reasonably possible),
• **Create** examples of rubrics for recognition/coauthorship and contribution/participation rules for open science, and
• **Publish** a paper describing the challenges discovered, lessons learned, advancements achieved, and how this work can be expanded upon.
Please make OSF/ORCiD accounts so I can add contributors!
Project Tour: HDRL’s HelioCloud

- Cloud computing environment
- Executable and shareable notebooks
- Large file storage supported via public s3 buckets
- Initial compute and storage costs funded by HDRL

https://daskhub.hsdcloud.org

...more tomorrow in S. Antunes’ presentation.
Project Tour: GitHub page

• Link to project webpage added to readme file.
• Scripts and notebooks stored in ‘DataWorkflows’ folder.
• Software environment information in top directory.

https://zenodo.org/badge/latestdoi/631044088
Project Tour: Linking It All Together

People on the OSF pages will browse the data in s3 buckets and the files on HelioCloud through an intuitive interface.

Contributors will perform all data analysis on HDRL’s HelioCloud.

People on the OSF pages will also see software, discussions, documentation and contributors embedded from GitHub.

Contributors will easily push/pull software between GitHub and HelioCloud.
Project Tour: Current Status

• Project posted on OSF with a DOI
• Dependency conflicts resolved on HelioCloud
• **Software runs on data in s3!** (except for SWMF GM outputs)
• Workflows being planned and developed
Project Tour: Path Forward

• **Work out kinks** in running PyHC software on data in s3 buckets.
• **Add Contributors!** *(Make an account on OSF/ORCiD so I can add you!)*
• **Link** HelioCloud, OSF and GitHub together (easier said than done!).
• **Streamline** workflows for contributors *(NEED TESTERS!)*.
• **Draft** contribution/participation rules based on JWST example.
• **Finalize** contribution/participation rules at Fall PyHC meeting.
• **Present** at AGU 2023 a (hopefully) ready environment.

Any burning questions?
Open Science and Heliophysics Infrastructure

- **Open Science** is the principle and practice of making research products and processes available to all, while respecting diverse cultures, maintaining security and privacy, and fostering collaborations, reproducibility and equity. ([https://www.whitehouse.gov/ostp/news-updates/2023/01/11/fact-sheet-biden-harris-administration-announces-new-actions-to-advance-open-and-equitable-research/](https://www.whitehouse.gov/ostp/news-updates/2023/01/11/fact-sheet-biden-harris-administration-announces-new-actions-to-advance-open-and-equitable-research/))

- **Why open science?**
  - Accelerates scientific discovery.
  - Greater collaboration and efficiency.
  - Enhanced transparency and reproducibility ([NASEM, 2018, p. 3](https://www.whitehouse.gov/ostp/news-updates/2023/01/11/fact-sheet-biden-harris-administration-announces-new-actions-to-advance-open-and-equitable-research/)).
  - Mandated by the U.S. White House and NASA.

- Image Credit: NASA TOPS ([https://zenodo.org/record/6565080#ZFPvCnbMKUk](https://zenodo.org/record/6565080#ZFPvCnbMKUk))
Open Science and Heliophysics Infrastructure

- **FAIR components** (*Findable, Accessible, Interoperable, Reusable*):
  - *Making good progress*: publications, observed data, metadata,
  - *Needs focused development*: modeled data, software,
  - *Exploration required*: model codes, software environments,
  - *The great unknown*: people, relationships, collaborations, ...

- **Reproducible results**
  Executable papers? Analysis environments? How long to maintain and what depth of reproducibility?

- **Open processes**
  How to perform science in the open from the beginning?

- **Inclusive collaborations**
  How to make collaborations open?

**FAIR data and open-source software are NOT enough!**
It is okay to start there, but we *must* look beyond for guidance on infrastructure design.
Observational Data

- A growing number of datasets...
  - Are **searchable** through a modern interface (using SPASE),
  - Have **citable** DOIs independent of publications,
  - Are **downloadable** both through web pages and APIs,
  - Are **browsable** via quick-look plots, and
  - Are available on the **cloud**.

**How can PyHC better advertise data access and analysis support in PyHC packages?**
Modeled Data

- Infrastructure supporting modeled data is far less developed.
  - No modern search interfaces,
  - DOIs are not assigned,
  - Few modeled datasets are downloadable through a website,
  - Only reduced versions are available through an API, and
  - Some quick-look plotting capabilities are available, but are not easily accessible.

How can PyHC help with these issues?
Sustained push is underway to **open-source all software** (including modeling code) generated with taxpayer dollars.

**Open-sourcing software is NOT enough.**
- Dependency conflicts (!),
- Conda/pip installability on multiple operating systems (e.g. Mac, Windows, Linux),
- Lacking documentation,
- Need examples and tutorials,
- Capability to run on the cloud,
- Maintenance for long-term reusability,
- Support staff for questions/problems, and
- Containerization for software environments?

*PyHC should take the lead here. What paths forward have low-hanging fruit?*
Where is this going?

• Build a **distributed data infrastructure** system:
  • Observational and modeled data hosted and served by multiple institutions,
  • Containerized model codes available on the cloud from multiple institutions,
  • All searchable from a united modern interface through **connected metadata**,
  • All accessible using multiple methods (e.g. file links, APIs, quick-look plots).

• Build a **collaborative analysis infrastructure** system:
  • Analysis environments with **software already installed** (and referenceable),
  • Reusable executable analysis tutorials for how to use the data,
  • Searchable through **connected metadata**,
  • Accessible through the cloud (e.g. downloadable containers or cloud platforms).

*How can PyHC prepare for, collaborate with, and enable these infrastructures?*
Discussion Time!
Discussion: PyHC support of Open Science
Discussion: PyHC support of Open Science

Scan the QR code or use the link to contribute to the discussion!
https://tinyurl.com/5n6tsxt6
What PyHC software changes are needed to better support the open science experience? Which of these need funding?

- How to run SWAP data stored in S3 buckets?
- Need a metrics script example that uses PlasmaPy
  - mean error, RMS root square mean, absolute error

We need funding for long-term maintenance of code!

DoI's for software - should PyHC offer DoI minting or should a DoI be required for software to be listed by PyHC?

PyHC packages should test against the main branch or release candidates of their dependencies

DoI minting or should a DoI be required for software to be listed by PyHC?

PyHC office hours?

Data archives point to PyHC packages that work well with that data.

How can PyHC better advertise data access and analysis support in PyHC packages?

- Generate a SPASE metadata record for software to better enable discoverability at MDMP?
- Create links on the archive dataset pages to relevant subtask?

- Also try to improve the PyHC gallery :)?
- Out-of-the-box: A custom GIP-powered sheetbot on pyDataFing that can explain what's available in conversation?
- Community crib sheets with examples for how to use a package - e.g., someone learns to use a new package, the sample code they create (with some curations!) could be useful examples for the next newcomer?

Webinars (similar to pySPEDAS)

- Have a standard set of tutorials that anyone going to a meeting can take so that we have lots of representatives who do the training

Advertisement at ALL conferences

- Can pull from the PyHC summer school tutorials, our gallery, etc and create some kind of nice pre-packaged set of examples for training?
- Hosting work parties to improve project documentation

Should PyHC hire a community manager like Astropy?
How can PyHC help with accessibility issues for modeled data?

- Provide guidance to community on adding cruces to other languages (e.g., C.C++, Fortran, DL, etc.)
- Advise on how to include scripts in other languages as dependencies in a pipeline installation.
- Make a specific call in the HTM NASA AO for improving model access
- Create metadata standard(s) for simulation inputs/outputs
- Standardized data storage formats for access methods
- Many modelers are using ParaView, so PyHC could spend resources to support ParaView.
- Simplify the process to have SpacePy as a dependency

Partner with IHDEA

How can PyHC prepare for, collaborate with, and enable the next generation of infrastructures?

- Provide containerized analysis tutorials
- Publish a referenceable containerized PyHC software environment
- Demonstrate how to run software on the cloud.
- Generate a DDI for software packages in PyHC as a service
- What software standards to pursue for I/O data access (e.g., Fati2, cfdpoint)
- Partner with CERCS to engage more with the community and find out more about what users need
- Continued outreach to the next generation of scientific software developers and scientists through engaging and informative PyHC summer schools, presentations, and booths at other meetings, etc.
- Is Podman/Docker a viable 'package system' for premissions to code to work?
- Docker alternative: https://podman.io/
- The Execution Model prevent arbitrary code execution (aka no bitcoin mining and spam)
- data access beyond local files of all kinds (HAPI, other companies' clouds)
- Hardware dependence support (large data, high core count, large memory, GPUs)
- Hackathon or similar to try out a lots of different ways to containerized/commeratalize a turnkey PyHC environment
- Allow multiple options for obtaining a viable collection of PyHC tooling
- Use singularity instead of Docker
Useful Links

- OSF project page: https://osf.io/v4drt/
- GitHub project page: https://github.com/rebeccaringuette/MagnetopauseExecutablePaper or https://zenodo.org/badge/latestdoi/631044088
- HelioCloud: https://daskhub.hsdcloud.org
- Polson et al. 2022 journal article: https://doi.org/10.3389/fspas.2022.977781
- Polson et al. 2022 on Deep Note: https://deepnote.com/workspace/shawn-polson-c095a0fb-0f2d-416d-9e94-c4a9c4e8e54d/project/PyHC-Paper-101b9646-3fd0-4978-a48e-a4f3e708a9ac/notebook/Making_anExecutable_Paper_with_the_Python_in_Heliophysics_Community_to_Foster_Open_Science_and_Improve_Reproducibility-c3a5772e5e24ce15426b01696d52251
- This presentation’s link on PyHC’s google drive: https://docs.google.com/presentation/d/1c2bP0zDdiJWMCPZ2zxm9U3z80NPH_SC/edit?usp=share_link&ouid=11819833928784120748&rlz=1C1Chbia
- Miro board link with preliminary project workflows (view): https://miro.com/welcomeonboard/Q0SNeGiZM0taVGkxkKjzKkzVzJsdzKud3R3SIF4RWJ4RGhONX8mazZheThhd1d4aUNLc1VDOFM1WHzXa01ZWXwzMdc0NDU3MzU3OTkO0DcyOTEyIDi=？share_link_id=953968385486
- Miro board link with PyHC 2023 spring session discussion (edit): https://miro.com/welcomeonboard/M2lRd2NNSXhPcXpPRXZ/dBFHcXcwRHNq ZyUzSk9XelBzeVdKT2p3NTHhQlVBRkG4VEJjaXc5MWNGT3dhUkxR3wzMdc0NDU3MzU3OTk00DcyOTEyIDi=？share_link_id=522570131865
Project Tour: Preliminary Project Workflows
Project Tour: Preliminary Project Workflows

Software Contribution Workflow

- User’s dir directory created by admin
- git clone
- Install working software environment
- User modifies package
- Changes ready for public
- Click button on HelloCloud to generate a pull request on Github
  - Select repo from drop-down list (e.g., main, fork, forks, ... fork)
  - User enters description of changes or PR (commit)
- Pull Request created on selected Github repo
- Github maintainer denies Pull Request
- Github maintainer accepts Pull Request

How to make this easy. Install in a Docker container? Other options?

Need a tutorial on this process.

Embed Github Interface or link to it?

User makes requested changes and resubmits PR

Email sent to user with feedback

New script pushed to HelloCloud site and public site

Embed Github Interface or link to it?

https://miro.com/app/board/uXjVMMaO61I=/?share_link_id=703942255605