J. Space Weather Space Clim. 2020, **10**, 12

https://doi.org/10.1051/swsc/2020011

JSWSC

Available online at:
www.swsc-journal.org

Topical Issue - Space Weather research in the Digital Age and across the full data lifecycle

TECHNICAL ARTICLE OPEN ACCESS

Reproducible Software Environment: a tool enabling computational reproducibility in geospace sciences and facilitating collaboration

Asti Bhatt*, Todd Valentic, Ashton Reimer, Leslie Lamarche, Pablo Reyes, and Russell Cosgrove

Center for Geospace Studies, SRI International, Menlo Park, 94025 CA, USA

Received 8 December 2019 / Accepted 4 March 2020

Abstract – The Reproducible Software Environment (Resen) is an open-source software tool enabling computationally reproducible scientific results in the geospace science community. Resen was developed as part of a larger project called the Integrated Geoscience Observatory (InGeO), which aims to help geospace researchers bring together diverse datasets from disparate instruments and data repositories, with software tools contributed by instrument providers and community members. The main goals of InGeO are to remove barriers in accessing, processing, and visualizing geospatially resolved data from multiple sources using methodologies and tools that are reproducible. The architecture of Resen combines two mainstream open source software tools, Docker and JupyterHub, to produce a software environment that not only facilitates computationally reproducible research results, but also facilitates effective collaboration among researchers. In this technical paper, we discuss some challenges for performing reproducible science and a potential solution via Resen, which is demonstrated using a case study of a geospace event. Finally we discuss how the usage of mainstream, open-source technologies seems to provide a sustainable path towards enabling reproducible science compared to proprietary and closed-source software.

Keywords: Geospace science / open source software / data collaboration / computational reproducibility

1 Motivation

The Sun-Earth system is complex and multi-dimensional in nature, requiring understanding of various interdependent regions and governing parameters. Geoscientists studying this system increasingly rely on diverse datasets from instruments that are analyzed together and often assimilated in a model to arrive at a scientific result. Geospace science is in the midst of a data revolution, with an increasingly growing number of continuous measurements of the geospace system than ever before (NASA alone is archiving more than 27.5 petabytes of data, see page 2 of EOSDIS (2018)), enabled by various ground- and space-based instruments worldwide. In order to fully exploit these data for new discoveries, a simultaneous revolution in accessing, processing, and reproducing research results for robust scientific productivity needs to happen (McGranaghan et al., 2017). While challenges and barriers to accessing and processing data still exist, in the current manuscript we will focus the discussion on computational reproducibility of geospace research results.

In general, an increased focus on scientific reproducibility has been developing in the geosciences with many scientists expressing doubts about whether the results in their field of research are reproducible (e.g., Baker, 2016). Reproducibility of results is a necessary condition of the scientific process, but is challenging in multiple aspects. While empirical and statistical reproducibility of scientific results are complex and challenging subjects, in the era of digital tools, computational reproducibility poses yet another challenge that can and must be addressed (e.g., Teytelman, 2018). Recently identified in the Open Source Software Policy Options for NASA Earth and Space Sciences (National Academies of Sciences, Engineering, and Medicine, 2018), many challenges for computational reproducibility are largely methodological; caused by a lack of standard practices in the way data and software are treated within the scientific community. Some of these challenges include: a lack of public repositories for diverse data, ownership and long term curation of data, sharing data and software openly and transparently with proper acknowledgment, and data and software version control.

1.1 Computational reproducibility

The challenge of computational reproducibility has two distinct, but related aspects to it:

Data transparency: Geospace research instruments produce more continuous data than ever before, with a high degree of variety. In order to achieve computational reproducibility, these

*Corresponding author: asti.bhatt@sri.com

data need to be discoverable and accessible, along with understandable metadata to begin with. While this is true for independent datasets, solving complex geospace problems means analyzing data from disparate instruments, which also require the data to be interoperable.

Software sharing: Computational reproducibility typically requires sharing the code written to arrive at a particular result. This can be challenging as user-written code often relies on local files and hard-coded file paths. Other challenges are the changes in computer ecosystems, software versions and obsolescence of features with time. These often make software codes utterly unusable, or give downright wrong results. In a recent example from biosciences, Bhandari Neupane et al. (2019) discovered a problem in how different operating systems treat different Python functions, calling into question hundreds of scientific studies using a widely-used code.

The Integrated Geoscience Observatory (InGeO) was envisioned as an open-source data-processing and computing environment in which open-source data-processing tools contributed by instrument providers facilitate the seamless mapping of observational data from diverse instruments to a common grid along with additional tools contributed by the broader scientific community. These tools support comparative analysis and interpretation, all coming together to create computationally reproducible research products. Initial work with InGeO included a prototype data-linking and software-sharing system as an integrative activity under the US National Science Foundation (NSF)'s EarthCube program to facilitate community-centered and data-driven systems science research with an initial focus on the geospace science community. This system morphed into a REproducible Software ENvironment (Resen) platform that uses mainstream digital technologies to create a community software platform for accessing diverse geospace data through community developed software tools to create computationally reproducible research results. While the use of proprietary software packages, such as MATLAB or IDL, is prevalent in the geospace community, Resen specifically depends only on open-source software. This allows researchers to distribute work done with Resen freely without concerns about licenses and legality of redistributing proprietary software packages and improves the ability of the computational environment to be preserved independent of outside companies continuing to support old versions of software. It also means that Resen is more accessible for researchers in the developing world that may not have access to expensive proprietary software. In recent times, several new initiatives have increased use of open-source software like Python in the geospace community (Stoneback et al., 2018, 2019). Resen is used to create results that preserve the underlying compute environment in addition to the data and lines of code to ensure that the code will run exactly and without modification, so the results can be shared with anyone freely. In this paper, we describe the usability of Resen via a case study and outline the enabling technologies.

2 REproducible Software ENvironment (Resen)

Resen has been developed to be used in two ways: (1) online (in the cloud), with a user authentication system and limited computational resources available to each user, and (2) installed on a user's own computer, with similar functionality as the online version, but with additional availability of computational power, storage space, and ability to import and export results for reproducibility. Currently, the online tool is envisioned to operate as a demonstration of the local tool, but the online tool has been developed such that it can be trivially scaled up as required.

Resen is an open source tool that encapsulates the data analysis process in a way that facilitates reproducible data analysis while simultaneously enabling users to access community developed software, libraries, and data (Integrated Geoscience Observatory Team, 2019a). Resen is built using mainstream and open source software products, combining Python and Docker to provide a containerized Jupyterlab interface (Docker Inc., 2019; Project Jupyter, 2019). This combination of software provides a highly portable and flexible data analysis environment while minimizing development and maintenance time, leading to longer term sustainability of the software platform. The means that Resen can be run on Linux, macOS, and Windows and scientific data analysis can be shared and reproduced across operating systems.

Within Resen, data analysis environments are referred to as buckets. Resen lets users create buckets, each of which is a self-contained Linux software environment where data analysis results can be easily reproduced. Under the hood, Resen buckets are simply Docker containers, with a pre-built Docker image, called *resen-core*, developed by the InGeO team (Integrated Geoscience Observatory Team, 2019b). The *resen-core* docker image is based on a standard Ubuntu Linux image and has been specially designed so that Resen buckets come pre-installed with a variety of commonly used geospace libraries and software packages (Fig. 1). This allows users to avoid the arduous and often time-consuming task of manually installing software on their local systems and instead get right to research. For example, several community Python packages have been developed in the geospace community, however, the installation of many of these packages is not a trivial exercise and many software packages require operating system specific dependencies that has often meant Windows users are unable to use them. Having standard or commonly used community software packages available in a containerized environment, provided by *resen-core*, significantly lowers the barrier to entry for many useful, but highly specialized community-developed software tools. In addition, Resen buckets contain software from various geospace data providers that make it easy to access and download their data from public repositories. Users can also mount in their own custom data sets to analyse or simply upload them via the web browser graphical user interface that is provided by a Jupyterlab server running in the docker container.

Containerization is widely adopted in industry for running web applications in controlled environments. Docker is a powerful and feature filled container software, however, Resen abstracts away many of the more complicated details of working with Docker containers to create a straight-forward user experience. Users need to install Docker, which is simple due to wide-spread usage. Additionally users need to install Python 3 and Resen to get started. An easy way to get started is to use the installation guide that is available on readthedocs (https://resen.readthedocs.io).

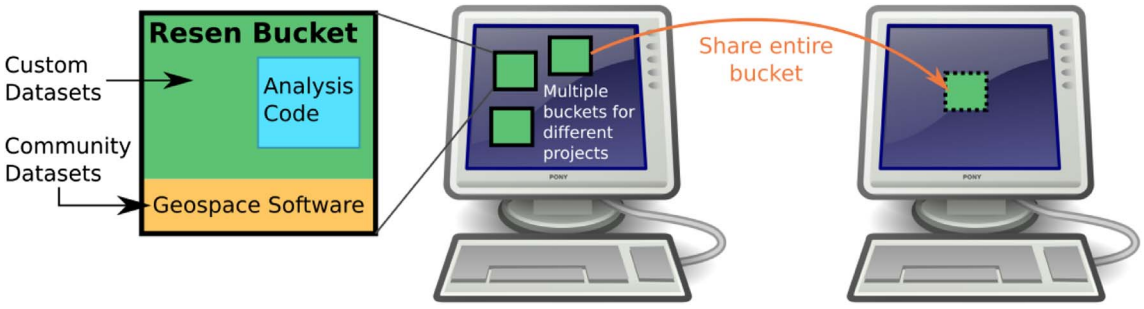

Fig. 1. Resen buckets are isolated environments where code can be developed and run on top of existing community-developed geospace software. Users can have multiple buckets at any given time, each configured for and holding code needed for a particular bucket. Then, entire bucket, complete with analysis code and any custom configuration, can be shared with collaborators so they can quickly and easily reproduce the entire analysis.

Currently, Resen provides a command line application (Table 1) that can be started by entering $ resen at a command prompt. In the Resen application, $create command walks the user through the steps to create a new Resen bucket. After a bucket is created, a JupyterLab server can be started and stopped using the start and stop commands. The JupyterLab server provides a graphical user interface via the user's preferred internet browser. With the JupyterLab interface, users can upload and run code, write new programs, create Jupyter notebooks of their work, and even install new software into the Resen bucket. This lets a user completely customize their Resen bucket and configure it however they wish.

Users are able to export Resen buckets, so that the buckets can be shared with collaborators. Exported buckets include all of the analysis code, results, and any custom installations or configurations or software packages that were performed by the user. In addition to sharing buckets with other Resen users, buckets may be uploaded to a research data repository, such as Zenodo (https://zenodo.org), where the bucket will be publicly available and citable with a DOI such that anyone may download, execute, and rapidly reproduce research results. Sharing buckets with collaborators will enable them to rapidly reproduce results and/or perform additional analysis without having to install dependencies or configure their own systems.

The online Resen system is designed to let users try out working in Python and Jupyter notebooks without having to install anything on their own computers. The online system allows users to install libraries through a command line interface, which is part of JupyterLab. Currently, the online system has limited resources and is only intended for demonstration purposes, however the online system was built using JupyterHub and readily scales on servers with a large number of users and plentiful system resources. Both the online and downloadable versions of Resen contain several tutorials to help new users to become familiarized with Python, Jupyter notebooks, and basic usage of some community software libraries. For using other Python libraries, users are able to consult the documentation and examples distributed freely on the web by library maintainers.

Finally, it is common in geospace science communities to also work with heritage codes developed in compiled languages such as FORTRAN or C, which require specialized libraries. Compiling and using these codes can prove to be significantly challenging for users, as it often requires using specific versions of compilers or system libraries, which may be unavailable or challenging to install on some operating systems. Since *resen-core* is built on a base Ubuntu Linux image, it is relatively simple to configure a Resen bucket such that heritage codes will compile and can be run easily. Furthermore, it is possible to develop Python wrappers to further simplify running these codes. A completed bucket could then include both a fully functional simulation code, visualization tools, and analysis tools, greatly simplifying the usage of the code for most users. Recall that Resen buckets are portable and can be run on Linux, macOS, and Windows, so Resen buckets containing heritage codes can be trivially operated in any of these operating systems.

To illustrate how one may use Resen to perform computationally reproducible research, the following section describes a case study wherein Resen is used to analyze data collected by several instruments that observed aurorally generated ionospheric disturbances at mid-latitudes.

Table 1. Resen commands.

create	Creates a new resen bucket from user input
list	Lists all available resen buckets and if they are running
status	Prints the status and details of a resen bucket
start	Starts a jupyter server in a resen bucket
stop	Stops a running jupyter server in a resen bucket
import	Import a bucket that someone else exported
export	Export a resen bucket for publication or sharing
remove	Removes an existing resen bucket
quit	Exit Resen

3 Case study using Resen

This case study describes using Resen to address a specific research question. It is common for research questions to arise from observations made by an individual measurement with further research requiring additional contextual information from several other instruments to understand a geophysical process. Below we outline a motivating research question and addressing it both with and without the usage of Resen.

3.1 Motivating research question

The Mid-latitude All-sky-imaging Network for Geophysical Observations (MANGO) measures ionospheric airglow at 630 nm across the continental United States. On November 7, 2015, the MANGO network observed a brightness wave propagating southwestward in two of the western imagers. This can be observed in the quick look movies at the MANGO network's own website https://mango.sri.com/data. By viewing the movie, the brightness wave appeared to display characteristics of a large-scale traveling ionospheric disturbance (LSTID), i.e. high speed and long horizontal wavelength covering the entire imager field-of-view. While LSTIDs have been observed for a long time, what makes this event interesting is the significant westward tilt of the LSTID phase front. LSTIDs have typically been defined by their east-west elongation and largely southward propagation (Shiokawa et al., 2002) or a small westward tilt up to 20° (Song et al., 2012). Source mechanisms behind LSTIDs observed at mid-latitudes are generally understood to be auroral zone processes like Joule heating, Lorentz forces, or energetic particle precipitation that launch acoustic gravity waves propagating equatorward (Song et al., 2012; Lyons et al., 2019). However, the exact process behind the gravity wave launch still remains unclear.

Recently, Lyons et al. (2019) did a correlation study of several LSTID events as observed in the north American sector with high-latitude magnetometer and auroral imagery to understand the driving mechanism behind LSTIDs observed in airglow measurements from MANGO imagers and the total electron content obtained via distributed Global Poisitioning System (GPS) receivers. One of the questions posed in the summary of Lyons et al. (2019) was, "Does the phase front orientation of LSTIDs depend upon the longitude relative to the longitude of the driving disturbance?" As the westward tilt of the LSTID phase front observed in the MANGO data was close to 45° on Nov 7, 2015, we decided to investigate the high-latitude drivers and background geophysical conditions that may explain it.

3.2 Methodology

The first step was to analyze the MANGO data to clearly identify characteristics of the observed LSTID. Then, since LSTIDs occur during geomagnetic unrest, we needed to look through geomagnetic indices like *Dst* and *Kp*, which are available from several providers including NASA's OMNI database. Next, as the event was captured at mid-latitudes in north America, we decided to look through data from other instruments with coverage in the continental United States. Key data that can shed light on an event like this are thermospheric winds, which are typically measured by Fabry-Perot Interferometers (FPIs). Similarly, velocity measurements from coherent scattering of irregularities in the ionosphere can also tell about the large-scale movement of plasma, e.g. from the SuperDARN radars. The next step would be to look at geomagnetic activity at high-latitude to understand the driving mechanism.

Some of the datasets like winds from FPI and GPS TEC measurements are available through the CEDAR Madrigal repository (Rideout, 2017) that provides an application program interface (API) to download the data in various programming languages including Python and MATLAB. Others are

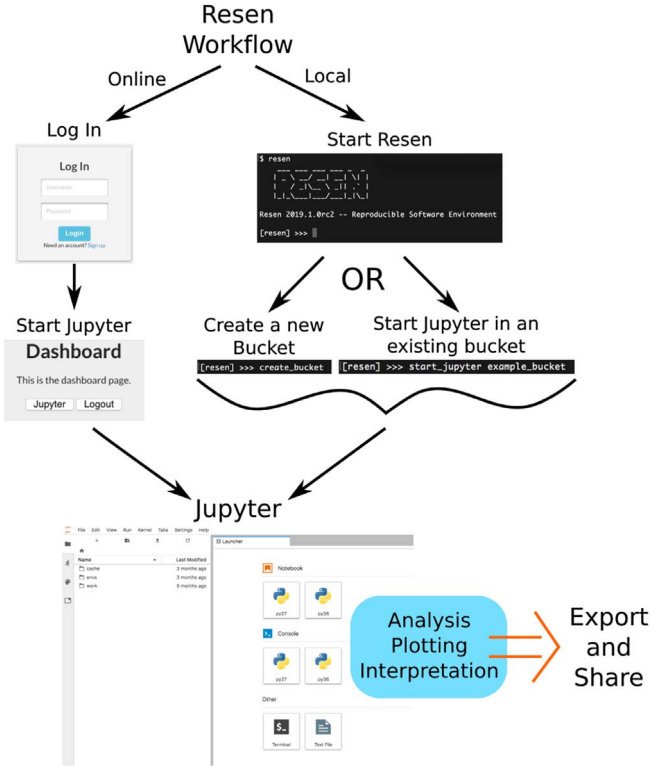

Fig. 2. Resen can be used both online through the InGeO website or locally to take advantage of local computing power.

available through individual data providers. The SuperDARN radar observations are available to view through a web interface hosted by the Virginia Technical University, and the data have historically been available to access and analyze using the DaViTPy Python library from the Super Dual Auroral Radar Network Software Team (2018). However, at the time of this analysis, the data accessing functionality of DaViTPy was unavailable. With this information, below we outline how the accessing and analysing these data would be done without Resen, and how we carried it out with Resen.

Without Resen: There are broadly two ways of accessing and analyzing all the relevant data without Resen – (1) going to the source of each dataset (person or website), obtain relevant files, understand the data and metadata, and write your own analysis and visualization routines in the language of your choice, or (2) installing various software tools developed by different data providers on your machine in the programming language they are made available in. Some of these may mean using proprietary software tools. In the last few years, several Python libraries have been developed within the geospace community to access and analyze some of the widely used data. In our example, data available from Madrigal can be accessed via the MadrigalWeb library; MANGO data are available through mangopy Python library; Spacepy is a Python library that can be used to access geophysical indices (Morley et al., 2011); DaViTPy and pydarn are developed by the SuperDARN team for analyzing SuperDARN data. In addition, several other Python libraries, such as cartopy (Met Office, 2010–2015), are required for additional analysis and plotting, and to satisfy dependencies of data analysis libraries. This would take

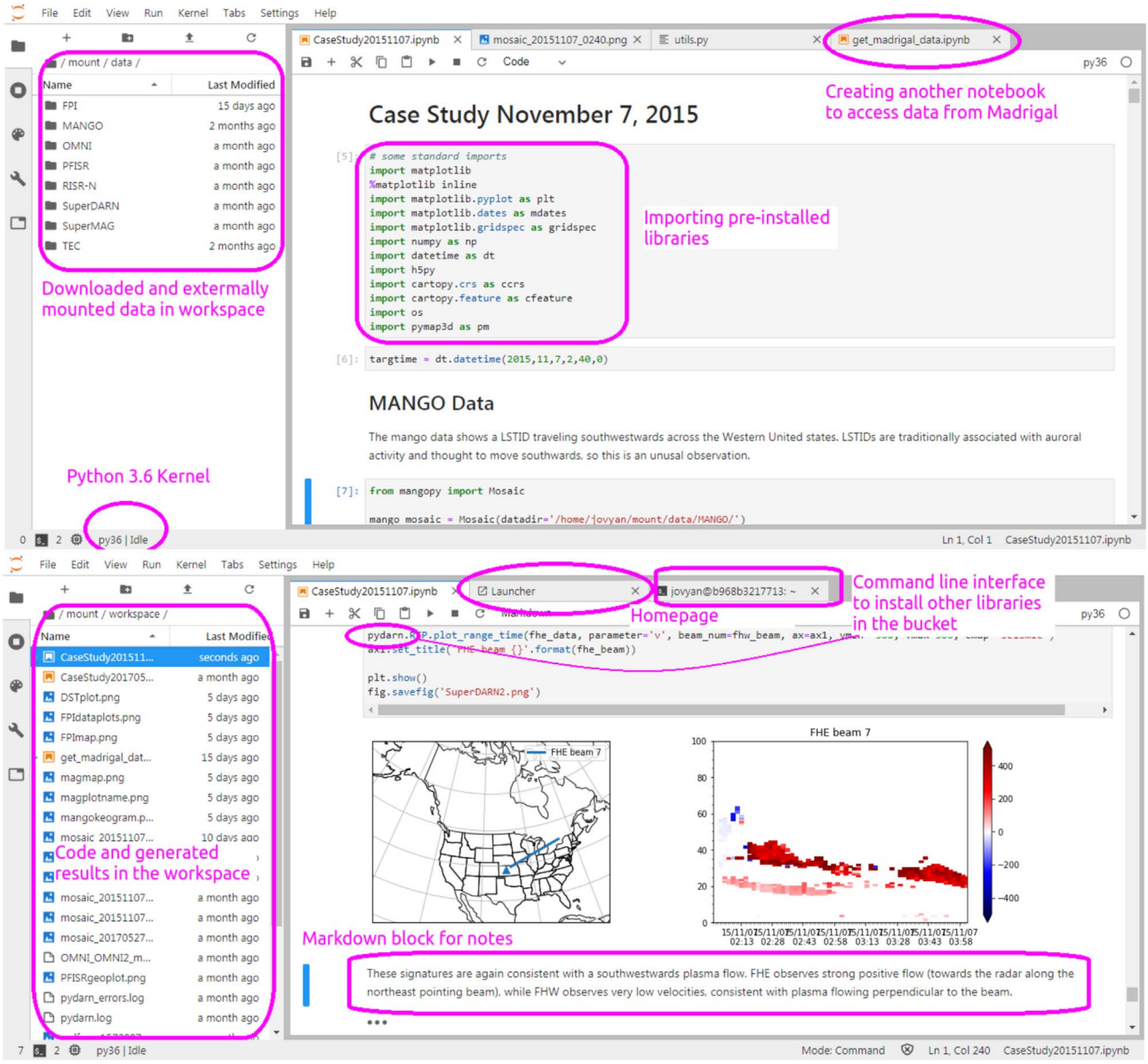

Fig. 3. Annotated screenshots from JupyterLab interface in Resen showing different components. The top panel shows the beginning of the case study notebook with data and importing pre-installed libraries. The bottom panel shows different functionalities of JupyterLab interface, where one can install other libraries like pydarn in the Resen bucket.

significant amount of effort on the part of the researchers. Once the research results are output, the researchers would have to manually bundle the data and software together in order to publish in a journal. These results would not be computationally reproducible without recording the underlying Python library versions and operating system dependencies.

With Resen: Using the workflow described in Figure 2, we created a new Resen bucket and mounted external data obtained from SuperDARN and SuperMAG in it. We used the available *resen-core* image that has all of the libraries mentioned above pre-installed with all the dependencies satisfied. The data files are readily available to analyze in the JupyterLab workspace opened from Resen. We used the mangopy library pre-installed in Resen to download and visualize the MANGO data and to do additional processing, like creating keograms (see Fig. 4). Similarly, we used the MadrigalWeb library within Resen to download the FPI and TEC data files, and spacepy to access the OMNI database for geophysical indices. While the DaViTPy library is pre-installed in Resen, we are given to understand that the SuperDARN team is developing a new Python library called pydarn (available from https://github.com/SuperDARN/pydarn, retrieved on Dec 7, 2019) for data visualization. It is not officially released at the time of writing this manuscript, hence is not pre-installed in Resen. However, the JupyterLab interface gives the flexibility to install other Python libraries through the command line interface, and we installed the pre-release version of pydarn in the Resen bucket for this case study to visualize SuperDARN data. Figure 3

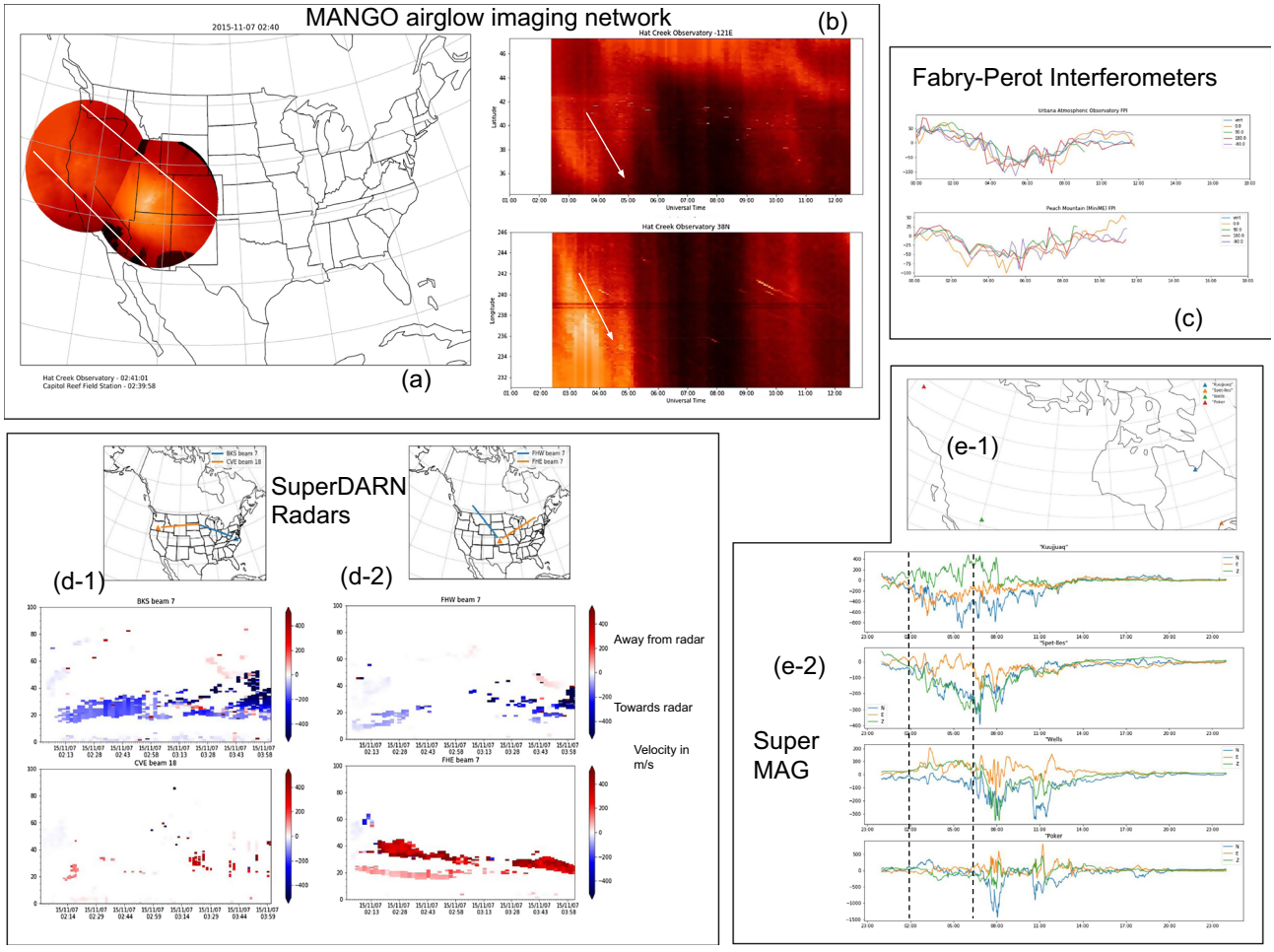

Fig. 4. A figure palette containing images made within Resen from diverse data sources. (a) MANGO mosaic image created using functionality of the mangopy library installed within *resen-core*. The image shows large enhanced airglow band tilted southwestward in two western imagers. (b) Keograms (north-south and east-west cuts at specific longitude and latitude respectively) created for the Hat Creek (western most) imager showing a signature of a southward and westward propagating LSTID from 0230 UT to 0400 UT, also indicated by arrows. (c) Two FPIs from Urbana and Peach Mountain in mid-eastern United States showing southwestward winds between 0200 UT and 0600 UT. 0° is north and 90° is east. The positive values always refer to northward or eastward. FPI data were downloaded using the madrigalweb library installed in *resen-core*. (d) SuperDARN radar velocities from four northern hemisphere radars. The beam directions are shown in the maps at the top of d-1 and d-2 panels. The Blackstone SuperDARN radar (BKS) beam 7 shows negative velocities, which are westward, and the Fort Hayes East (FHE) beam 7 shows positive velocities that are southwestward. SuperDARN data acquisition was done outside of Resen, but analysis used the Python library DaViTPy developed by the SuperDARN consortium. (e) Four magnetometer sites from the SuperMAG network show the magnetometer fluctuations indicating high geomagnetic activity, which seems to progress from north to south and east to west, as indicated by two dashed lines. SuperMAG data were downloaded separately, and uploaded to the Resen bucket.

shows two panels showing screen grabs from the JupyterLab interface part of Resen with various functionalities available to the users.

3.3 Results

The mosaic figure with MANGO data (Fig. 4a) shows the pronounced westward tilt, and the north-south and east-west cuts taken at specific longitude and latitude (keograms) show both the southward and westward propagation between 0230 UT and 0400 UT. Looking at the available FPI data on Madrigal, we couldn't find any thermospheric wind measurements co-located with the MANGO imagers. However, since the LSTID is a large-scale phenomenon, we looked at wind measurements from Urbana (40.13° N, 271.8° W) (Makela, 2015a) and Peach Mountain (42.27° N, 276.25° W) (Makela, 2015b), FPIs in the eastern United States. Between 0200 UT and 0430 UT, the observed winds from both of these instruments show south and westward propagation. We looked at SuperDARN velocities in four radar beams, all oriented differently. The Blackstone and Fort Hayes East radar beams show strong westward/southwestward velocities. Fort Hayes West radar beam shows velocities that are close to zero as the orientation of the radar beam indicates that any perpendicular flows should not have a strong directionality. Through Madrigal, we only had access to the TEC data from the world-wide GPS

receiver network, which doesn't allow for a clear detection of a TID. Deriving differential TEC requires specialized processing (e.g., Lyons et al., 2019) that is beyond the scope of this work. So we did not gain any significant information from the TEC data. Now looking at the source region for the LSTID, the magnetometer data at high latitude show that strong fluctuations in the geomagnetic field occur at sites northern and eastern prior to western and southern. The vertical lines in panel e-2 in Figure 4 show that fluctuations in the North geomagnetic component at the 'Wells' station, which is south and west of Kuujuak, and west of Spet-lles, occur hours later than similar fluctuations at these two stations. Similarly Spet-lles displays a delay compared to Kuujuak. From this analysis, we learned that the pronounced southwestward tilt in the LSTID seemed directly correlated with the fact that the geomagnetic activity progressed from northeast to southwest in the source region as well. This analysis brings up another question with respect to the propagation of LSTIDs – *Is the energy deposition process that launches the gravity wave in the auroral zone a single large event, or a progression of one or more events that determines the propagation direction of the launched gravity wave eventually manifesting into an LSTID?*

The Resen bucket with containerized data, analysis codes, and underlying library installations is available to download at https://zenodo.org/record/3564835#.Xeyt21lKjCI for other researchers to look through this work, and reach their own conclusions. Since this bucket automatically contains the versions of software products needed to produce the research results, no further documentation is necessary to outline the dependencies.

4 Summary and future work

A reproducible software environment developed within the EarthCube funded US National Science Foundation project provides a space for geospace community software to be hosted seamlessly, while enabling the broader geospace research community to make use of available software and data from several providers, and to create computationally reproducible results. Two key advantages of this platform are: (a) it eliminates the need to install several diverse Python libraries on users' own local machine, the effort behind which can be significant, and (b) it enables creating containerized results that can be shared and reproduced by other researchers seamlessly. We demonstrated using the Resen platform through a case study pertaining to propagation direction of an LSTID, which originates in the high-latitude during energetic processes. Our multi-instrument analysis uses diverse data from disparate geospace instruments, and several community software libraries. The results indicate that propagation direction of the geomagnetic activity at high-latitudes may play a role in determining the propagation direction of the LSTIDs.

Future work for Resen includes adding functionality to access data from various satellite missions by installing stable versions of relevant Python libraries developed within the geospace community, as well as working with various stakeholders from the community and soliciting feedback on the usability of Resen. This will be accomplished via dedicated sessions and direct interaction with users during community conferences, including virtual workshops specifically designed for students. Additionally, collaboration with another EarthCube funded initiative has already begun, which will ultimately result in the inclusion of a Python version of the widely-used assimilative mapping of ionospheric electrodnamics (AMIE) model within Resen (Matsuo, 2020).

Finally, as we enter the FAIR (Findable, Accessible, Interoperable, Reproducible) data era (Stall et al., 2018), containerized results show a path forward for sharing reproducible results that are immutable with changes in underlying software tools. We hope that Resen tool becomes a resource for the geospace community worldwide that not only provides access to a wide array of geospace Python libraries, but also a way to advance open access to scientific data and software.

Acknowledgements. This work was supported by the US National Science Foundation grants ICER-1541057 and CSSI-1835573 to SRI International. The software and data used for the case study (Fig. 4) have been published on Zenodo, DOI: https://doi.org/10.5281/zenodo.3564835. The editor thanks two anonymous reviewers for their assistance in evaluating this paper.

References

Baker M. 2016. Is there a reproducibility crisis? *Nature* **533**: 452–454. https://doi.org/10.1038/533452.

Bhandari Neupane J, Neupane RP, Luo Y, Yoshida WY, Sun R, Williams PG. 2019. Characterization of Leptazolines A–D, polar oxazolines from the Cyanobacterium *Leptolyngbya* sp., reveals a glitch with the "Willoughby-Hoye" scripts for calculating NMR chemical shifts. *Org Lett* **21(20)**: 8449–8453. PMID: 31591889. https://doi.org/10.1021/acs.orglett.9b03216.

Docker Inc. 2019. *Docker*. Available from https://docs.docker.com/install/ (Last accessed 6 December 2019).

EOSDIS N. 2018. *NASA Earth Science Data Systems Program Highlights 2018*. https://cdn.earthdata.nasa.gov/conduit/upload/11232/ESDISHighlights.pdf.

Integrated Geoscience Observatory Team. 2019a. *Resen*. Version 2019.1.0rc2. Available from https://github.com/EarthCubeInGeo/resen (Last accessed 6 December 2019).

Integrated Geoscience Observatory Team. 2019b. *Resen-core*. Version 2019.1.0rc2. Available from https://github.com/EarthCubeInGeo/resen-core (Last accessed 6 December 2019)

Lyons LR, Nishimura Y, Zhang SR, Coster AJ, Bhatt A, Kendall E, Deng Y. 2019. Identification of auroral zone activity driving large? Scale traveling ionospheric disturbances. *J Geophys Res: Space Phys* **124**: 700–714. https://doi.org/10.1029/2018JA025980.

Makela J. 2015a. *Data from the CEDAR Madrigal database*, CEDAR Madrigal Database. Available from https://w3id.org/cedar?experiment_list=experiments/2015/uao/06nov15&file_list=minime05_uao_20151106.cedar.002.hdf5 (Last accessed 25 November 2019).

Makela J. 2015b. *Data from the CEDAR Madrigal database*, CEDAR Madrigal Database. Available from https://w3id.org/cedar?experiment_list=experiments/2015/ann/06nov15&file_list=minime08_ann_20151106.cedar.003.hdf5 (Last accessed 25 November 2019).

Matsuo T. 2020. *Recent progress on inverse and data assimilation procedure for high-latitude ionospheric electrodynamics*, Springer

International Publishing, Cham, pp. 219–232. https://doi.org/10.1007/978-3-030-26732-2.

McGranaghan RM, Bhatt A, Matsuo T, Mannucci AJ, Semeter JL, Datta-Barua S. 2017. Ushering in a new frontier in geospace through data science. *J Geophys Res: Space Phys* **122(12)**: 12586–12590. https://doi.org/10.1002/2017JA024835.

Met Office. 2010–2015. *Cartopy: A cartographic python library with a Matplotlib interface*, Exeter, Devon, UK. http://scitools.org.uk/cartopy.

Morley SK, Koller J, Welling DT, Larsen BA, Henderson MG, Niehof JT. 2011. Spacepy – A Python-based library of tools for the space sciences. In *Proceedings of the 9th Python in Science Conference (SciPy 2010), Austin, TX*.

National Academies of Sciences, Engineering, and Medicine. 2018. *Open Source Software Policy Options for NASA Earth and Space Sciences*, The National Academies Press, Washington, DC. https://doi.org/10.17226/25217.

Project Jupyter. 2019. *Jupyter Lab*. Version 0.35.5. Available from https://jupyter.org/install (Last accessed 6 December 2019).

Rideout W. 2017. *MadrigalWeb*. Version 3.1.11. Available from https://pypi.org/project/madrigalWeb/ (Last accessed 6 December 2019).

Shiokawa K, Otsuka Y, Ogawa T, Balan N, Igarashi K, Ridley AJ, Knipp DJ, Saito A, Yumoto K. 2002. A large-scale traveling ionospheric disturbance during the magnetic storm of 15 September 1999. *J Geophys Res: Space Phys* **107(A6)**: 1088. https://doi.org/10.1029/2001JA000245.

Song Q, Ding F, Wan W, Ning B, Liu L. 2012. Global propagation features of large-scale traveling ionospheric disturbances during the magnetic storm of 7–10 November 2004. *Ann Geophys* **30**: 683–694. https://doi.org/10.5194/angeo-30-683-2012.

Stall S, Yarmey LR, Boehm R, Cousijn H, Cruse P, et al. 2018. Advancing FAIR data in Earth, space, and environmental science. *Eos* **99**. https://doi.org/10.1029/2018EO109301. Published on 05 November 2018.

Stoneback R, Klenzing J, Burrell A, Spence C, Depew M, et al. 2019. *pysat/pysat v2.1 (Version v2.1)*. Zenodo. https://doi.org/10.5281/zenodo.3546270.

Stoneback RA, Burrell AG, Klenzing J, Depew MD. 2018. PYSAT: Python Satellite Data Analysis Toolkit. *J Geophys Res: Space Phys* **123(6)**: 5271–5283. https://doi.org/10.1029/2018JA025297.

Super Dual Auroral Radar Network Software Team. 2018. *Data and Visualization Toolkit-Python*. Version 0.8. Available from https://github.com/vtsuperdarn/davitpy (Last accessed 6 December 2019).

Teytelman L. 2018. No more excuses for non-reproducible methods. *Nature* **560**: 411. https://doi.org/10.1038/d41586-018-06008-w.

Cite this article as: Bhatt A, Valentic T, Reimer A, Lamarche L, Reyes P, et al. 2020. Reproducible Software Environment: a tool enabling computational reproducibility in geospace sciences and facilitating collaboration. *J. Space Weather Space Clim.* **10**, 12.