Dear Editor,

We are indeed grateful for the feedback provided by the reviewers, as well as all the volunteers that make Seismica possible. Seismica's focus on community collaboration and open access aligns so well with the ethos of open-source software that it was an obvious choice for this work, and we are glad the reviewers agree.

We have copied the reviews below and addressed each concern/question in blue. Some of the suggestions for additional features/functionality are in our development plans, but due to the magnitude of some of these tasks we were not able to address all of them in the immediate future. We did, however, open several issues to help track and prioritize these requests.

In addition to the reviewers' comments, we have added several lines in the acknowledgement section to credit other python packages DASCore uses that we initially overlooked.

Reviewer D

Thank you for sharing this well-written manuscript with me. In the manuscript, the authors are addressing a very important issue that there requires an open-source, community maintained, easy-to-use Python library for Distributed Acoustic Sensing (DAS). As ObsPy becoming the de facto python library for seismology, the authors introduced DASCore, which is designed for common operations including IO, visualizing, analyzing, and managing of DAS data.

According to this succinct manuscript, DASCore provides various levels of support for data IO, visualization methods, and other transformation functions. The entire package and associated functionalities are based on two new data structures, **Spool** and **Patch**, which are innovatively designed by the authors and explained in Figure 1. There are in-line script snippets with small sharable dataset for testing, which I think are good practice for first-time users. Although still at its early version, I can see from the project GitHub repository that it's been actively maintained with all sorts of good open-source practice (Continuous Integration/Continuous Deployment, code coverage test, documentations, etc.).

In summary, I believe this manuscript is at a good shape and is a great fit for the **Seismica software report** section.

Questions

1. In Line 69, it is said that "a simple HDF5-based index" exists to speed-up data query. Can you explain what information is indexed and if indexes can be shared with other users?

We added a few lines clarifying these details.

2. In Line 142, the authors state that there is a tentative plan to manage metadata with DASCore. What would be the reason hindering this manuscript version of DASCore to implement this?

This is a great question, and clearly a needed feature. In short, the reason it doesn't exist yet is the magnitude of the task. The implementation needs in-depth knowledge of various existing standards and initiatives, such as ProdML and Earthscope's DAS metadata reporting format that is undergoing further development (https://github.com/DAS-RCN/DAS_metadata). To be truly effective as a reference implementation, it will require significant collaborations with interested parties, such as EarthScope and the DAS interrogator manufacturers. Therefore, we chose to defer this significant task to a future work, which will probably merit its own publication.

3. For software development, test data tends to be small but widely covered. Did the authors implement any scaling test on larger datasets (GB~TBs), which I feel necessary and may reveal any hidden overhead.

Yes, we are using DASCore for datasets up to ~50 TB with ~230k files. We have added some information about this to the manuscript at line 74. DASCore's indexing is based on the implementation in ObsPlus, which the NIOSH affiliated authors actively use to manage waveform archives with millions of files. Because of this experience we are confident it can scale to several million DAS files, which, based on the file size, could easily be in the 100s of TB range.

4. Would DASCore be available to process cloud-hosted DAS data, e.g., Ridgecrest DAS from SCEDC? Similarly, would DASCore be useful to access data from PubDAS (Spica et al., 2023)?

Better cloud support is something we are currently thinking about. HDF5 files don't work very well in cloud environments due to concurrency issues, but currently it is the most popular format to store DAS data. We plan to add support for more cloud friendly formats like TileDB (#155) and Zarr, and have engaged in discussions with the Earthscope SAGE Data Management Center staff to plan for future support, but this is still down the road. We do support converting DASCore Patches to and from Xarray data structures, however, which provide mature cloud support. Some datasets from PubDAS can be supported through DASCore, but there are also some datasets within PubDAS that deviate from standard DAS I/O formats due to the lack of required standard submission formats in the PubDAS repository. PubDAS has 2 years of remaining funding support to continue offering data housed at University of Michigan, so it is currently a temporary open archive. The DASCore contributor team includes part of the PubDAS team, so as these data are migrated to long-term archives in the future and undergo standardization, we will be able to appropriately support these data.

Reported Issues

5. I tried following the tutorials on GitHub and run the DASCore on one of my own DAS data. During installation, conda is not able to resolve and install the package well (empty environment). pip can install the library though.

We have opened an issue (#357). Several developers tested various operating systems and we didn't find any issues with the Conda installation. We would invite the reviewers to open an issue with more specific info about their OS, system architecture, and python version so we can better address this issue.

6. For some specific cases, DASCore is not able to index the data folder. For example, a permission denied error is raised if the folder contains another subdirectory which the user does not has permission on.

I suppose this is a somewhat unusual case we have not encountered yet. We have opened a new issue (#358).

Reviewer E:

While DAS technology has seen widespread use across various fields of seismology, the lack of a unified software solution for data processing has been an issue. The emergence of this Python package comes at a crucial juncture, offering a well-designed solution to streamline DAS-based studies. I commend the team for their dedication to addressing this need.

Having extensively tried many functions within this package, I can attest to its user-friendly nature and impressive feature set. The incorporation of concepts like 'patch' and 'spool,' inspired by the popular Python package 'obspy,' is particularly commendable. These structures will undoubtedly resonate with 'obspy' users and enhance their experience. Additionally, the support for units and related operations is innovative and valuable, especially given the diverse range of physical variables that different interrogators measure.

However, I encountered an installation issue initially, as the version installed via 'conda' (dascore-0.0.13) resulted in numerous error reports when running scripts.

This is unfortunate, and we have opened an issue to investigate (#357). It is interesting conda did not install the latest version (DASCore 0.1.0). It is possible the reviewer attempted to install DASCore into an existing environment which had dependency conflicts with the latest DASCore version.

After some debugging, I found that reinstalling using 'pip' resolved the issue (dascore-0.1.0).

Furthermore, while the current version's 'processing' module offers useful functionality, it is somewhat limited in scope.

We currently have several additional processing features on our planning board (https://github.com/orgs/DASDAE/projects/2/views/1) but feedback from users and reviewers as to which features they would like to see can help us properly prioritize these efforts.

Additionally, the package currently only supports either 'read' or 'write' operations for most data formats. While I understand the authors' intent to promote the 'DASDAE' format, it would be beneficial to address these gaps and enable support for both reading and writing operations.

Currently, DASDAE format directly emulates the Patch data structure, and so is the only format that provides lossless conversion. We have thought about SEGY (with extended headers) and PRODML write support as well, but have not yet encountered a need to prioritize this effort.

Looking ahead, I eagerly anticipate the expansion of the package's functionality, making it an even more powerful tool for researchers in the field. Overall, the manuscript accompanying the package is concise and well-written, and I recommend its publication in its current form.