Coordinating the Next Generation of Spectroscopic Processing and Analysis Tools (13-16 Nov 2023, Tucson, AZ): Workshop Report

Workshop Rationale

Following The Future of Astrophysical Data Infrastructure meeting held earlier this year, the goals going into the workshop were to (1) survey the current landscape of available tools for processing and analyzing spectroscopic data, (2) identify areas of need and opportunities for collaboration, and (3) make first practical steps toward building these tools together. Our program included plenary discussions and breakout working sessions, and ended with an extensive discussion of long-term collaboration/development strategies and planning for future follow-up meetings.

Outcomes

A brief summary of the primary outcomes of the workshop are as follows:

- **Platform Consensus**: The attendees reached the consensus that collaborative development of community-focused spectroscopic reduction and analysis tools should happen within the Astropy-coordinated packages `specutils` and `specreduce`.
- **Long-term collaboration strategies**: The attendees will follow two main avenues for continued collaboration on projects initiated, or seeded, by this workshop: (1) remote Zoom meetings held once per month for development check-ins and collaborative coding sessions and (2) hold a follow-up workshop next year following the ADASS meeting in Malta with a similar format: informal discussion with most of the time allocated to practical implementation of spectroscopic tools within these two packages.
- **Building a common terminology**: We began a *living document*[1] used to collect and clarify terms that are needed to make sure development proceeds with use of a common lexicon. This document will continue to grow, and we welcome input from our larger developer community. Editing/acceptance protocols are TBD.
- **Completion of small-scale work packages**: Attendees developed and worked on numerous small-scale projects that ranged from tackling long-standing issues in `specutils`, making small steps toward improving the functionality of `specutils` (particularly `Spectrum1D` I/O methods), performing brief feasibility studies, establishing infrastructure for longer-term comparisons between different data-reduction pipelines,

[1] GitHub pull request that includes this document is still pending. The link provided is where the document *will* appear. We will remove this footnote once the the PR has been cleared.

and discussing more experimental approaches to data reduction (machine learning or AI driven, full forward modeling).

Table of Contents

Discussion Summaries

Extensive notes were taken throughout the workshop, both in plenary sessions and during each breakout. These are linked to at the end of the document. Here we summarize many of the discussions held.

Common Ground & Shared Values

To ensure that all the attendees were on the same page, we started by discussing shared values and terminology (see the next section).

- It's important to emphasize that our shared goal is to facilitate good science, both by workshop attendees and their collaborators as well as the larger community. A major way we do that is by providing open-source tools that are user-friendly and lower the barrier to entry, both technically and economically.
- We value intentional design that facilitates interoperability and open development to yield robust, modular, and maintainable software and, hence, long-term stability.
- We stand on the shoulders of decades of algorithm and tool development. It's important to recognize this by collaborating with each other and, by proxy, previous generations using existing algorithms. Specifically, we want to avoid reinvention of existing

algorithms and minimize duplicated effort in their implementation or translation to new coding languages.

- This development can be done while also recognizing that not all development is general, and bespoke, purpose-built software must be welcomed and encouraged, particularly when it drives frontier algorithm development.
- We value bold ideas that may, at first, seem intractable or daunting, particular within the limitations of our own work schedules, with the knowledge that these often motivate more incremental progress toward more idealized solutions.

Common Terminology

Given the variety of expertise and experience among the workshop attendees, we spent substantial time building and discussing a common terminology. This grew rapidly, and we felt it worthwhile to maintain the result as a living document, hosted by either astropy or its coordinated packages, `specutils` and `specreduce`. The result of the original discussion is included in the running notes (see links at the end of this document), and inclusion in one of these repositories is pending.

Hack Summaries

A core part of the workshop were "hack" sessions where attendees pitched ideas for small-scale breakout discussions, code development, or feasibility studies. Not all pitches were acted on; attendees worked together on the ideas they were interested in. All pitch ideas are collected in the workshop slides, and many of the discussions maintained extensive notes (see the relevant links). The following provides summaries of most of the ideas with detailed notes/progress:

Long-term `specreduce` Development Planning

A few ideas centered on scoping and planning for future developments of `specreduce`. These were combined into a series of discussions. First, the group reviewed a `specreduce` workflow concept from previous astropy meetings and discussed these within the context of BANZAI, DRAGONS, and pypeit. Looking at the `specreduce` code, we found that the code had moved on in some ways since the workflow diagram, and we noted some inconsistencies that could be easily addressed. Generally, the data-reduction pipelines (DRPs) represented in the room had a consistent workflow, but the details were not discussed deeply. Second, the group conceived of a framework for testing/comparing similar algorithms between the existing DRPs. These ideas were developed into a testing module within `specreduce`, and an initial module was implemented that will eventually enable testing the BANZAI, DRAGONS, and pypeit wavelength calibration algorithms. These will be expanded for other common data-reduction tasks, like tracing, sky-subtraction, object extraction, etc, over the coming year (on a best-effort basis). Third, we recognized the need for a common/standardized data format. This was largely discussed in the context of the testing apparatus (i.e., the common interface needed for the wavelength calibration modules from each DRP), but we also recognized the need for common output formats. On the user side, this is largely within the context of wrappers that enable users to read relevant outputs into `specutils` objects.

Short-term development of `specutils`/`specreduce`

A number of the hack ideas led to immediate improvements to `specutils` and a number of pull requests and GitHub issues. The following is a short list of these:

- **`Spectrum1D` I/O**: The "tabular-fits" writer for `Spectrum1D` seen as inadequate because it doesn't maintain header data. Group found that header data is propagated if in HDU1, but not if in HDU0. A plan for a new reader/writer were discussed and a draft PR with tests demonstrating that header keywords are not in the expected location was issued.
- **`specutils` Plotting:** The group revived an existing PR that adds basic/rough plotting functionality in spectutils (e.g., something like `Spectrum1D.plot()`). They found that use of the `NDCube.plot()` function had undesirable behavior, whereas the bespoke `plot_quick()` function can be exactly as desired. Future work will see this through.
- **Treatment of NaN values in `specreduce`:** NaN values in any wavelength channel knock out the entire channel for all spectra (in the cross-dispersed direction). This was discussed and a solution was proposed.
- **B-spline Sky Subtraction:** Improved b-spline sky subtraction following Kelson (2003) was implemented using scipy's b-spline fitter and a pull-request was issued. Initial performance looks very good.

Novel data-reduction techniques

A few of the hack ideas focused on novel data-reduction techniques, namely the use of machine learning and artificial intelligence or full-fledged forward modeling. In terms of ML/AI, the general consensus was that this was an approach that we should keep in mind, but it's not going to prove to be a short-term solution. A preferred approach would be to replace individual data-reduction steps (like wavelength calibration) with ML/AI modules and slowly build toward a full ML/AI implementation. Training sets will be crucial, particularly those that are relevant to the full scope of instrument use and how the instrument response evolves over its lifetime.

Forward modeling approaches to data reduction exist; the group discussed ynot and dLux in particular (which includes ML-powered modules). These approaches can be incredibly powerful (and are likely the way of the future), but they need to be computationally tractable. Typical ("inverse model") approaches of current DRPs will continue to be faster. The group also discussed the power of forward modeling in providing simulations (digital twins) of data that are invaluable to instrument development, DRP development, and planning observing programs. If these digital twins are realistic enough, they can also be used to generate simulated data for training AI/ML-based data-reduction models.

Handling Survey-level/Big Data Use Cases

Enabling processing and analysis of very large data sets from (and between) upcoming/existing surveys is a critical component of present-day astronomy research. This group discussed important considerations for `specutils` (the `Spectrum1D` object in particular) that meet this need. Specifically, group members wanted functionality that enabled access to not just the final calibrated science spectrum, but also any best-fitting model, sky-only spectrum, etc, that is currently not enabled by the `Spectrum1D` object. More generally, each survey will likely have its own idiosyncrasies that should both be accommodated but not hinder joint analysis with other surveys. The group discussed different approaches that could be implemented in `Spectrum1D` that address this. They also proposed convening members of SDSS, DESI, Euclid, SPHEREx, and other surveys to discuss adoption of and modifications to the IVOA spectrum datamodel needed to help facilitate survey science.

Lightning Talks

Five-minute lightning talks were given on the following topics, largely introducing existing software packages as a point of reference useful to our discussions:

- C. Shanahan (STScI): Jdaviz
- S. Juneau (NOIRLab): SPARCL
- T. Pickering (UA/Steward): `specreduce`
- K. Westfall (UCO): pypeit
- B. Cherinka, J. Sanchez-Gallego (STScI): Marvin
- C. Simpson (NOIRLab): DRAGONS
- C. McCully (Las Cumbres): BANZAI
- B. Weaver (NOIRLab): SPARCL + Jdaviz
- S. Bailey (LBNL): nearly_nmf
- R. Pucha (UA): Emission-line fitting with Astropy modeling

Development Platform Consensus

The general consensus was that the astropy-coordinated packages `specutils` and `specreduce` should be the shared development space. The primary goal for these packages is to develop a network of interoperable tools. Workshop attendees were encouraged to become maintainers of these packages to accelerate development progress. "Maintainers" do not necessarily need write privileges; responding to issues and reviewing code solely via GitHub is extremely valuable. Incentive structures are needed to encourage contributions to these packages. Development of these structures is an ongoing effort. The following options were discussed at the workshop: developing and maintaining good developer documentation, sponsoring hack events at meetings (AAS, ADASS, etc.), and encouraging imperfect contributions by supporting necessary refinements.

Development goals for `specreduce`, in particular, are for it to be a dependency of many/most specialized DRPs. I.e., DRPs are built using underlying `specreduce` functions/modules. One way to achieve this is by having current pipelines contribute (or “upstream”) low-level algorithms to `specreduce`. Ideally, these packages would then use these components of `specreduce` (minimizing code reproduction and the possibility of diverging implementations), but this isn't necessarily a requirement given the burden this places on existing pipelines. The short-term goal is for `specreduce` to be a collection of tools/algorithms for developers of future DRPs. The longer term goal is for `specreduce` to be a more broadly accessible and complete toolbox, but not necessarily an application (i.e., something that can be called from a terminal command line). An important consideration for the long-term viability and sustainability of `specreduce` is to implement functionality that can be run on both an individual laptop and large-scale HPC/GPU/??? clusters, either locally or in the cloud. This requires performance metrics/benchmarks and scaling estimates; however, this should be secondary to early, best-effort implementations (i.e., beware of premature optimization). Ultimately, `specreduce` should include a cookbook/tutorial, similar to the document P. Massey produced for IRAF, The User's Guide to Reducing Slit Spectra with IRAF; inspiration can also be drawn from the CCD Data Reduction Guide.

Facilitating Contributions

Generally, workshop attendees expected it would be difficult to allocate time for direct contributions to `specutils` and `specreduce` within the scope of their normal workload. This is largely because each person's home institution has its own priorities, such as direct contributions to their specialized DRPs. E.g., most investment in software development comes from funding of particular projects or science programs, which cannot necessarily devote effort toward more general development. However, it is worth emphasizing that each attendee's home institution were willing to commit their time for the workshop, which is a positive sign for the value placed on open-source community-driven development of generally accessible tools. Limited capacity to commit to development of `specutils`/`specreduce` is expected to be an ongoing concern, but a few possible solutions were discussed. **Under the model of `specreduce` becoming a core dependency of existing DRPs, work on `specreduce` effectively enters the scope of working on those specialized packages, once the up-front effort of migrating functionality to `specreduce` has been done.** Work on that up-front effort was the topic of one of the hack sessions, and there is motivation to continue that work. Also, managers generally appreciate the need for something like ~20% of a developer's time for exploring the field; i.e., understanding new things coming online, how the industry is changing, taking advantage of advances in technology, etc. This can also be toward upstreaming algorithms to `specreduce`. Finally, contributions to `specutils` and `specreduce` can be *directly* supported by astropy, at the level of small fractions of an FTE. Proposals for targeted contributions to `specutils`/`specreduce` are welcome. We should also explore opportunities to include international collaborators and funding sources.

Follow-up Meetings and Development Efforts

Workshop attendees planned to continue their development activities/hacks over the coming year and made a tentative plan to meet again next year following the ADASS meeting in Malta, but avoid overlapping with the IVOA meeting similar to this meeting. Other opportunities for ad hoc meetings include the upcoming AAS meeting (Jan 2024, New Orleans), the Astropy

Coordination Meeting (spring 2024, Netherlands), and the astronomy-focused SPIE meeting (June 2024, Japan).

Aside from these in-person meetings, we plan to hold monthly Zoom meetings to continue development efforts started during the workshop, on a best-effort basis. These will be organized over the ORCA Slack workspace in the `#spectroscopy` channel. We plan to summarize progress from these meetings via addenda to this document.

Relevant links

- The list of attendees is here.
- The workshop schedule is here.
- Slides used to guide the discussion during the workshop are here.
- Extensive running notes were kept here, which have branches to many notes subsets.
- A list of possible projects are here, which led to most/all of the hacks discussed in the slide deck.