Grisms Roadmap (AstroGrism Project)

Point of contact: Ivelina Momcheva Megan Sosey

Preamble

(Focus here is on multi-object observations; exo-planet observations focused on a single target should be considered separately, that community feels they are doing fine.)

Grisms provide a unique observing mode used in many unique scientific application. We are currently supporting two instruments with grisms (ACS and WFC3) and two more are coming on JWST (NIRCam and NIRISS), for a total of four instruments with grisms in the near future. NICMOS also had grisms, however due to complexities of the instrument these data will not be considered. Furthermore, WFIRST will have grism spectroscopy. Whether or not that is supported in any way at STScI remains to be seen, but showing leadership in grism spectroscopy in needed for our current suite of instruments.

Data reduction and analysis for grism observations is much more complex than for typical long-slit or multi-slit spectroscopy. The challenges are numerous and will not be elaborated on here but suffice to say that reduction, analysis and visualization techniques developed for long-slit and multi-slit spectroscopy are not generally applicable to grism spectra. That said, successful reduction and analysis algorithms and libraries do exist and the challenges are not insurmountable.

Historically, STScI inherited the aXe package from European Coordinating Facility and has been supporting it for the several years. While the aXe package does a decent job for a very basic extraction, it has been challenging to maintain and expand. Furthermore, running the package is quite complicated, installation is frequently challenging and it still does not do analysis (e.g., line and redshift fitting). The associated axe2web and axesim packages share the same issues. As a result of the limited support of the software itself and the lack of in-house development, grism expertise within SCSB and DATB is limited (to two people, Megan Sosey and Howard Bushouse).

Expertise however does exist in the INS teams (WFC3, ACS, NIRCam, NIRISS, WFIRST) and in fact a large fraction of the really deep grism expertise is currently at the institute. (Short of the software developed for Euclid.) There are three major teams that have done significant work on grism data and they all have members amongst current staff. The first are the team members who supported the data reduction and analysis of the PEARS, GRAPES, and FIGS data (large grism programs with ACS and WFC3): Nor Pirzkal, Russell Ryan, and Nimish Hathi. In the early phases, these programs heavily relied on aXe, but current analyses focus on LINEAR, EM2D, and other custom tools. The second is affiliated with the 3D-HST survey which developed a completely independent pipeline with Gabriel Brammer being almost a sole developer (threedhst, unicorn libraries; grizli was not used for 3D-HST but was developed by Gabe after 3D-HST was completed). Ivelina Momcheva was the project manager and Camila Pacifici was a member of that team. Finally, the WISP team, a pure parallel program using WFC3 also handled a large amount of grism observations, reducing them with aXe exclusively. Marc Rafelski and Alaina Henry are members of that team.

The volume of grism observations is currently quite low even though several large programs have been executed in the past to great success (3D-HST, GRAPES, PEARS, etc.). It is not clear if the demand for grism observations is low because users are unsure they can plan for and extract scientific value from the data (chicken and egg problem) or just interest is low and limited to a certain group of people in the know. But even if current rate of accepted proposals is low, the archive is quite rich and supporting archival research is a worthy cause.

With so much expertise concentrated at ST and no ongoing outside large grism project, it is highly unlikely that an outside group will develop a competing grism library. Note that the WFIRST grism software is currently expected to be created by IPAC and reuse much of the Euclid software (even though the observing strategies differ significantly for WFIRST). However, the Euclid pipeline is not open source and it is unlikely that the WFIRST pipeline will be either. Therefore, these developments do not satisfy the requirements for user-facing tools.

High Level Goals

1. Provide easy-to-use, well maintained, and well documented tools for doing multi-wavelength slitless spectroscopy related to grism capabilities of our missions (HST, JWST and WFIRST)
    a. First priority use is instrument calibrations - outputs need to be comparable and consistent with current tools to ensure continuity of callibrations
    b. Second priority use is user data reduction and analysis
    c. Third priority use is simulations for HST, JWST (maybe WFIRST)
    d. Easy to understand API for any libraries that are developed is above GUI development (GUI development is not at all a priority)
2. Reduce long-term maintenance costs by coordinating community libraries with similar functionality - i.e., there should be a single library with as few dependencies as possible that we maintain for grism support
    a. Including cross-functionality with astropy, i.e, this should be an astropy affiliated package
    b. Consider if running as part of pipeline operations is feasible and what are the requirements for that
3. Standardize the model objects used to describe and work with slitless spectroscopy grism data
    a. cross-functionality with specutils/Spectrum1D should be considered
    b. cross-functionality with spacetelescope/specviz for 1D extracted spectra should be considered
4. Standardize common, shareable files that relate setup information or transformations
    a. This may be a great place to use and promote ASDF
    b. Can be mission ~~/ survey~~ specific
5. Finish the move of aXe out of the PyRAF environment to make it only dependent on Python and it's current C libraries
6. Provide interoperability with the Jupyter ecosystem

Issues/Gaps/Decisions needed

Q: What is the problem here?

We have a number of different libraries that have been developed in the last few years, but none are have unit and regession tests, documentation and are clean and maintainable and in Python. Only aXe is maintained and supported by the institute.

Q: What is the shared functionality between aXe, grizli, Linear, Gsim/grismconf, and the JWST pipeline code?

A: One possible approach is to consider a line between pipeline code which carries out most basic pixel level operations and satisfies basic requirements and post-pipeline tools such as aXe, grizli, Linear.
There tools can be seen as "data analysis tools" and as such are not beholden to the same requirements that the pipeline is. They can pick up where the pipeline leaves off, preferably at approximately the same spot for HST (say FLT/FLC) and JWST (whatever level that is, 2?). The overlapping functionality is that they all need to use the same configuration files and ideally how these configuration files are interpreted.

Note that the HST and JWST pipelines currently differ significantly in their data treatment: HST only does basic pixel level calibrations while the JWST pipeline creates the trace models and customizes them for the observation, 1d/2d extraction are also performed, and the resulting WCS object is able to directly relate wavelength with pixel on the detector or sky location.

Q: What common libraries are each of the currently-being-used/developed tools relying upon? What community-only supported libraries do they rely on? Is spacetelescope already supporting the development of any of these packages?

A: The relevant statement to make here is that whatever is being developed should be open source and take advantage of open source tools. In an ideal world, it should be an astropy affiliated package.

Q: Who will be the final audience for created libraries/toolsets?

A: Instrument teams, community users

Q: How flexible should the initial products be? What is the minimum viable toolset that relates success. What functionality does it have?

A: There is a subset of functionality, possibly found in all the currently available grism codes that makes up the minimum viable product. Having a package with common core functionality and data interfaces is the central goal, additional functionality/packages that build upon that core package to enable and enhance specific scientific research would be the next step.

An exhaustive set of science use cases within the community needs to be collected. Notionally, there are many such use cases although the software functionality, processing steps and data products are likely to have substantial functionality.

1. Preprocessing steps specific for grism data (flat fielding, sky subtraction, alignment of direct image)
2. extracting a spectrum for any RA, Dec projected into a grism frame
3. accurate, iterative contamination model which takes into account the 2D light distribution of the source, the SED/slope of the spectrum and emission lines
4. allows for coaddition of spectra from different exposures (interlacing/drizzling) and spectra from different roll angles (fine, 1D extraction too)
5. redshift fitting including photometry
6. Create 2D emission line maps
7. Can be used to simulate a grism image based on a direct image and some input on SEDs, etc.

Q: What packages are the community currently using for grism analysis? Is there a way to see the breakdown of use/features easily? How often do users give up on using the known tools and write software of their own; is this due to installation, use, or lack of documentation/user support? if so, resources can be used to help further develop already existing packages. Do the current tools lack any desired functionality?

A: There isn't a single way to gauge what the community is using but generally the name of the game in grism data is who do you know who can reduce grism data. The aXe community can be found in the Help Desk e-mails. For the other packages, search ADS for the names of the developers and see how many unrelated survey/project papers they are on. People don't write their own software, because it's too complicated. And it's easier to send an e-mail. The 3D-HST release in a way opened the field for many (there are >1000 users) because you don't need to have friends at ST who can reduce the data for you.

A broader input on missing and desired functionality needs to be done vis-a-vis the science use cases.

Q: Can aXe be fully retired or fully replaced by current tools? Will aXe be deprecated by the new tools, by EOL for HST, or at some other fixed date?

A: TBD.

Q: Are there reasons why multiple approaches to the analysis of grism data makes keeping multiple packages around desirable? Does any common functionality exists in this case? Linear's use of forward modeling is an example of a fundamentally different approach.

A: We currently don't have a good understanding of the pros and cons of different approaches. Since not all code is OS, there really has not been a "bake-off" between different approaches. And ultimately, these differences are likely to be second order. There should be at least one proper package we support. That should be the goal. Multiple approaches have value but having multiple poorly supported approaches has negative value.

Q:What funding sources exist to support development, documentation, and users of any newly created tools. Are any of the current tools limited to mission specific functionality or funding?

A: HST, JWST, ~~WFIRST,~~ DDRF, AR are all available sources of funding to one extend or another.

Q: Are any of the current tools using open source practices but are not open development? Do any of the tools have licensing that would conflict with STScI software. Are any of the tools completely closed source?

A: LINEAR is open source available on https://github.com/Russell-Ryan/pyLINEAR, with basic CI implemented. IsEM2D open source/open development? The Euclid pipeline is completely closed source. Many tools developed by the community are completely closed source.

Q: What are the operating system and dependency requirements for each of the tools. How does this affect their future support and development within the community.

A: We no longer support IDL and IRAF software.

Q: Are there any software considerations that need to be considered for UV/optical vs IR grisms?

A: The WFC3/UVIS grism is more complex, any basic reduction tools need to be able to handle this in order to maintain calibrations. The HeII background emission is only in the IR.

Q: How do world coordinate systems factor into any of the current tools? What support packages for WCS are they using?(AstLib/pyast /starlink, Kapteyn, wcstools, astropy.wcs, gwcs)

A: Grizli is a mix of stwcs, and astropy.wcs with a couple of gwcs uses thrown in for a good measure. LINEAR uses astropy.wcs exclusively for the WCS libraries. EM2D developer can comment on their integration with WCS libraries.

Q: Does a grism package need to be based on spec-utils?

A: If it makes sense, yes, but grism spectra are sufficiently different that it might not work. Where applicable, like 1D extractions, definitely.

Q: Should post analysis of extracted spectra be included in the grism libraries or shared with other libraries.

A: Yes, see above.

Q: Should there be backwards support for reading/using old data formats? From multiple missions? From an explicit mission?

A: There are not many (any?) external missions that need to be supported so yes, but maybe not as far back as NICMOS. The benefit of OS is that people can extend it to their own data if we don't support it.

Q: Are flux units handled by the package?

A: Absolutely, fluxing should be part of the functionality.

Q: How should meta-information about the processed data be handled and saved?

A: TBD, depends on file formats.

Q:Are there any immediate functional needs for specific missions or the community?

A: The most immediate needs are HST and JWST.

Q: Define reasonable timelines for completion of basic functionality, then desired functionality.

A: This should be a 1-year project.

Known Packages that may be Directly Affected

There are currently four separate contingents working on grism related software, they have varying focus and methodologies:

aXe: Developed over a long timescale with ESA + STScI support, the original author is Grisms Roadmap (AstroGrism Project). It current only runs in PyRAF, but work is in progress to make it a Python+C package. It also only supports HST grisms.

threedhst: Developed by Gabriel Brammer for the use of the 3D-HST program. No longer maintained. Descriptions in Brammer (2012) and Momcheva (2016). Python code, no longer maintained. Written to work with 3D-HST data specifically and not easy to generalize to arbitrary data and multiple roll angles.

grizli: Developed by Gabriel Brammer for the use of the Grizli AR program, but has been extended to work with JWST and other missions. Approach is different from threedhst, meant to fit any grism observation including multiple roll angles. It exists as a python package.

linear: Developed by Russell Ryan, Stefano Casertano, Norbert Pirzkal. Unlike most grism extraction codes, LINEAR determines the optimal spectrum (for each source) from the entire canon of available data (that may be rolled or dithered). Has been tested to work with WFC3-IR, WFC3-UVIS, and WFIRST.

jwst: none of these are complete packages

- pipeline: The jwst pipeline contains software to extract grism spectra from niriss and nircam observations
  - see http://www.stsci.edu/hst/wfc3/documents/ISRs/WFC3-2017-01.pdf for the generalized grism dispersion solutions used in JWST and Gsim
- mirage: To simulate wide field slitless spectroscopy (WFSS) data it uses:
  - NIRCam_Gsim: to disperse imaging data
  - GRISM_NIRCAM: NIRCam-specific grism configuration and sensitivity files
  - GRISMCONF: grism dispersion polynomials

Package	Repository	Spacetelescope directed/owned

axe axe2web axesim	https://github.com/spacetelescope/aXe_c_code	x
axe-cookbook	https://github.com/spacetelescope/aXe_Cookbook	x
iraf/stsdas-anaysis/slitless - iraf based	https://trac.stsci.edu/ssb/stsci_python/browser/stsdas/trunk/stsdas/pkg/analysis/slitless	x
jwst (wide-field slitless mode code, uses grismconf updated polynomial design)	https://github.com/spacetelescope/jwst/tree/master/jwst	x
grizli: Grism redshift & line analysis software for space-based slitless spectroscopy threedhst (https://github.com/gbrammer/threedhst) unicorn (https://github.com/gbrammer/unicorn)	https://github.com/gbrammer/grizli/tree/master/grizli	
grismconf and Gsim (related to axe, with updated polynomial design) EM2D	https://github.com/npirzkal/GRISMCONF, https://github.com/npirzkal/NIRCAM_Gsim https://arxiv.org/pdf/1806.01787.pdf	
Linear: determines optimal spectra for extraction	https://github.com/Russell-Ryan/pyLINEAR https://ui.adsabs.harvard.edu/abs/2018PASP..130c4501R/abstract	x
mirage: simulate WFSS data for JWST (nircam) but does not actually do dispersion, only imaging	https://github.com/spacetelescope/mirage	
Out of scope: ~~WAYNE: a WFC3 IR Grism Spectroscopy simulator described in Varley & Tsiaras (2015).~~ ~~It can simulate both staring and spatial scanning observations.~~ ~~Currently the code is not public but requests for access can be sent to the author.~~	~~https://arxiv.org/abs/1511.09108~~	
Out of scope: ~~UVOTPY: Swift UVOT grism spectral extraction software~~	~~https://github.com/PaulKuin/uvotpy~~	

Stakeholders

Data Analysis Tools Branch	Science Platforms Branch[1]	DMD	DSMO	JWST	HST	WFIRST	INS	Community	Other
Larry Bradley	Megan Sosey Harry Ferguson	DATB SCSB SPB		NIRCAM NIRISS	ACS WFC3	Megan Sosey IPAC (SSC)	Norbert Pirzkal Swara Ravindranath Stefano Casertano Russell Ryan	IPAC (SSC-WFIRST) they are working with CNES in France astropy (for library consolidation, resulting affiliated packages) ~~3DHST in that grizli is their support package~~ Any large science teams using aXe for HST data analysis Any team that wants to do grism reduction	

1: The new Science Platforms Branch is effective March 16, 2020 when its new branch leads starts work.

Major Work Items

It would be preferable to first collate as much information as possible about the packages described above, which address the questions in the Issues/Gaps section so that focus can be placed on the best path forward, see table below:

Task	Priority	Difficulty	Progress	Depends On
Remove PyRAF dependency from aXe, upgrade python, packaging, and dependencies	High	Hard	In-progress	Megan Sosey availability
Examine current packages: understand differences in approach determine unique features beyond set described above and triage importance find dependencies and requirements	Medium	Medium		Ivelina Momcheva
Agree on high level goals, minimum functionality, stretch goals (e.g., WCS), priorities.	High	Medium		A group of stakeholders
Discuss refactoring of grizli with Gabriel Brammer	High	High		
Identify ST staff who can work on this effort, ideally people with ~50% availability	High	Medium		Harry Ferguson Ivelina Momcheva
Decide on overall library architecture requirements, API, I/O, class structure, testing infrastructure, in-code documentation, dependencies				
Scope and make a plan for work.		Hard		
If existing funding is not commensurate with effort, explore additional funding streams	Currently low	Medium		Ivelina Momcheva
Create documentation and tutorials to support users.	Medium	Low		available test data, their location, and libraries

Out of Scope

1. Providing support for HST instrument other than HST/WFC3 and HST/ACS up front (further additions possible)
2. Writing WFIRST specific tools (beyond existing functionality) unless funding for that moves from GSDC to STScI
3. Creating any web UI (outside of what's compatible with the jupyter ecosystem)
4. Direct updates to external-to-spacetelescope licensed and managed packages that aren't following open development methodologies
5. Spectral data calibration at the detector pixel level (Iva: there are certain operations such as sky subtraction and flat fielding that are different from imaging and not executed in pipelines for grism data so some detector level calibration must be done by grism package)
6. Starting from scratch
7. Translating major pieces of code into Python but implementing algorithms can be

Major Risks

1. No available personnel to organize or perform the actual work
2. No community buy-in for a coordinated library
3. Insufficient funding to achieve high-level goals
4. Lack of collaboration within the institute

Possible Funding Sources

1. No funding for data analysis tools have been contracted for WFIRST, the Grism Science Data Center is expected to do all calibration work and extraction for any grism data. However, it's likely that any tools produced by STSCI could be utilized by the team there.
2. STScI - CSI project for any improvements to the astropy core library or creation of affiliated package for grism analysis
3. STScI - JWST mission funding for extraction of current JWST pipeline code into a core library
4. STScI - HST mission funding for movement of HST specific analysis tools, such as aXe, into a core library
5. STScI - DDRF proposal to design and build grism spectroscopy community package, in conjunction with JWST and HST money for adapting current JWST or HST tools/pipeline
6. Donated effort by the community
7. GSDC (WFIRST) contributed code
8. AR proposal(s), e.g., GRIZLI

DSMO	HST	JWST	WFIRST	DDRF/JDF	Other
☑	☑	☑	☐	☑	☑ Grizli AR grant

Possible Labor Resources

DATB	DMD	INS	Community	Other
Megan Sosey		Norbert Pirzkal	Gabriel Brammer	
		Russell Ryan	GLASS collaboration	
		Swara Ravindranath	CLEAR collaboration	
		Nimish Hathi	WISPS collaboration	
		Jules Fowler	astropy developers (as freely contributed or through contract)	
		Matthew Bourque	GSDC or WFIRST SITs?	
		Catherine Martlin		
		Kevin Volk		
		Stefano Cassertano		
		Marc Rafelski		
		Alaina Henry		
		Camila Pacifici		
		Sware Ravindranath		
		(~2.5 FTEs in INS total)		

Resources:

2019-10-15 Meeting notes on WFSS in JWST pipeline (shared by Anton Koekemoer):

https://outerspace.stsci.edu/x/44ceAw

JWST WFSS algorithms discussion (shared by Anton Koekemoer):

https://outerspace.stsci.edu/x/2obHAg

Presentation my Marc Rafelski :