AstroGrism Minimum Viable Product

This is a page to record decisions on what is in or out for a Minimum Viable Project.

VOTING

Please provide your assessment of importance or difficulty of items on the MVP list. Steps for doing this:

1. Go to the MVP voting link provided in email or slack. Email Harry Ferguson for the link if you can't find it.
2. Watch the video or follow the steps below.
    a. Duplicate the sheet entitled **ScoringTemplate**
    b. Rename it to your name
    c. Expand each category using the **+** sign on the lefthand side
    d. Enter your ratings for **Importance** and **Difficulty** using the dropdown menus (stars for importance, gears for difficulty)
    e. Blanks are interpreted as the lowest priority. Delete your entry if you wish to leave a blank.
    f. That's it.

mvp_inst...ions.mov

The purpose of a Minimum Viable Product should get a practical, extensible, maintainable, well-documented product into the hands of users as fast as possible. There is a delicate balance between what is too minimal to be particularly useful and what is too ambitious to be viable. As a worksheet we can begin to construct a table of features/capabilities and categorize them in terms of importance to the user and viability for an early release based largely on existing code and/or calibrations and/or reference files.

MVP worksheet

Feature /capability	**Importance for MVP (must, should, nice)**	**Difficulty to deliver in MVP (high, medium, low)**	**Comments**
user-friendly APIs	should	high	This is at least for a quick/dirty run. This might be similar to what Nimish Hathi mentioned. This concern is also including how data and processed data are encapsulated in variables inside the working environment (e.g., Jupyter). For example, considering the aXe outputs (check this ticket in Jira for some info ⚠ ASTROGRISM-45 - Jira project doesn't exist or you don't have permission to view it.), there are many files produced and this is not obvious how to access these saved outputs. It would be nice if we have a wrapper for reading these outputs back into the working environment, and we design how a user can access mostly used information easily such as x.trace, x.wavelength, x.flux.
modularity	must	medium	This would be useful especially when we think of extendability either adding new grism definition from different facilities, or user customized objects from base classes.
Interactive GUI	nice	high	Interactive graphic user interface will help users to quickly examine visually.

documentation	must	medium	This is important, as we all can agree 🙂 Mitchell Revalski : "...detailed documentation, and reference information for what choices are best for tunable parameters (e.g. if you allow the user to fit a Gaussian, Voigt, or Lorentz profile to something, suggest common choices and pros/cons for different scenarios). Finally, pointing out common pit-falls and sanity checks that should be performed along the way are helpful."
minimal code comment standard	should	low	We should discuss about what would be the minimal requirement for code commenting. Have these requirements noted down, and make sure that all codes complied to the minimal standard before accepting any push to the main code body. We might assign someone specifically to go through all codes and check for the compliance.
Tutorial / cookbook / template	must	low	This should explain itself how it import it is. Mitchell Revalski : Jupyter notebook is suggested. Megan Sosey : Create notebooks, using a common example, for all the code bases that we have • To understand differences • To understand commonalities • To flush out better interface design for a common library product
minimal functionalities in prototype	must	medium	Kornpob Bhirombhakdi : At minimum, the prototype should be able to take inputs = {a grism image or a set of dithered images, other association files... such as direct image, background image, etc.}, and give outputs. There should be APIs for users to easily re-run with different parameters (as compared to aXe, all outputs will be in OUTPUT folder, a user has to manage saving this folder separately in a different name, and re-run almost the whole codes just to produce outputs with different parameters. From my aXe experiences, I think extracting grism spectra composes of i) locating a spectrum, ii) compute trace and wavelength given a spectrum, iii) extract (in aXe sense, this is the SPC files), iv) post-extraction calibration (e.g., aperture correction, combining spectra with outlier detection and rejection algorithm in case use avoid performing extraction on a drizzled image, or flux scaled to photometric points... aXe does not perform these steps. Mitchell Revalski : "...a working end-to-end reduction pipeline example designed for a straightforward set of observations (e.g. image-spectrum-image). functionalities in a prototype might include: 1) initialize code directories and packages, 2) identify and download grism observations in an Astroquery style, 3) perform minimal data-quality checks with automated modules and print a user report, 4) proceed to spectral extraction by identifying a spectrum (or taking user-input locations from direct imaging), calculate required trace, wavelength, and so on, 5) extract the spectrum and perform calibrations, 6) run basic sanity checks on the extracted spectra and provide user report, 7) plot the extracted data products and report on extraction parameters such as aperture, contamination estimates, etc." Mitchell Revalski : "...emission (and absorption) line finding, possibly using user-input spectral templates, and integration with a line-fitting routine to produce emission line maps, kinematic maps, and an output format that can easily be manipulated to produce user-desired diagnostics such as using the measured line properties to calculate densities, temperatures, abundances, reddening, general line ratios, and so forth." Mitchell Revalski : "One aspect that I've found helpful integrating into my own codes: allow for all tabular data (including list of figures) to be output in LaTeX ready format for tabular or aastex style deluxetables (and figure calls)."
Identify associated data sets			e.g. Find and download direct and dispersed images that overlap on the sky via an archive query
Data model for grism data in general			Megan Sosey : Create a data model for grism data in general, show it's use
Organization and bookkeeping			Conventions for file formats (in and out) metadata in files, file names, directory structure, output files (e.g. column names and units)
Geometric transformations			Outline all of the variants and what the use cases are (e.g. elaborate from Nor's presentation)
Astrometric registration			Align dithered observations
Simulations			Create a simulated 2D dispersed spectrum from a 1D spectrum and image morphology
Background subtraction			What are the different background components & approaches to estimating/subtracting for HST instruments?
Flatfielding			This can be subtle; the same approach can't be used in all circumstances. Maybe multiple user stories are needed?
1D extraction with no model assumptions			i) locating a spectrum, ii) compute trace and wavelength given a spectrum, iii) extract (in aXe sense, this is the SPC files)
1D extraction relying on SED models			At least for the contaminants: Forward-model a "reasonable" assumption for the spectrum (flat, polynomial, or SED from a template library; informed by the direct image photometry). Lots of variants here. Is it sufficient for the MVP to **enable** this without providing a rich suite of models or templates?
1D "optimal" extraction			Weighting data by the cross-dispersion profile

2D spectral extraction			Maybe multiple stories with different approaches to getting a 2D extracted dispersed spectrum?
Co-adding			Maybe several stories with different approaches to co-adding spectra taken at different orientations?
Converting counts to flux			Applying instrument throughput calibrations.
Aperture corrections			1. point sources based on PSF 2. morphology based on direct image
Outlier detection & rejection from multiple exposures			
Find an isolated emission line			EM2D use case Mitchell Revalski : "...emission (and absorption) line finding, possibly using user-input spectral templates, and integration with a line-fitting routine to produce emission line maps, kinematic maps, and an output format that can easily be manipulated to produce user-desired diagnostics such as using the measured line properties to calculate densities, temperatures, abundances, reddening, general line ratios, and so forth."
Create an emission-line map			Create a 2D emission-line map from spectra taken at different orientations
Fit a set of templates			Varying flux and redshift
Visual outputs			Simply plots, etc. Mitchell Revalski : ...."plot the extracted data products and report on extraction parameters such as aperture, contamination estimates, etc."
Integration to LaTeX			Mitchell Revalski : "One aspect that I've found helpful integrating into my own codes: allow for all tabular data (including list of figures) to be output in LaTeX ready format for tabular or aastex style deluxetables (and figure calls)."

Mindmap (work in progress)

Email Harry Ferguson for editable link.

Style & Standards

Coding and documentation will follow the STScI style guide.