



STScI | SPACE TELESCOPE
SCIENCE INSTITUTE

EXPANDING THE FRONTIERS OF SPACE ASTRONOMY

JWST Data Analysis Tool Evolution September 2019-March 2021

Harry Ferguson, DATB Branch Manager
on behalf of the JWST Data Analysis Development Forum (JDADF)



Takeaway messages

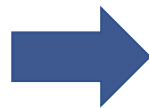
- Community input on JWST data-analysis tools is welcome at all levels from prioritization to open-source code collaboration
- Jupyter notebooks will play a big role in our data-analysis support
- We are changing visualization technology to enhance re-use between desktop, notebooks, and remote workflows



Data Analysis Tools

Post-pipeline analysis tools for STScI's missions and the scientific community

Post-Pipeline Data



Data Analysis Tools
 Read/Write
 Inspect
 Clean
 Transform
 Visualize
 Detect
 Measure
 Model
 Document

Results, Knowledge, Publications



A CANDIDATE $z \sim 10$ GALAXY STRONGLY LENSED INTO A SPATIALLY RESOLVED ARC

BRETT SALTON¹, DAS COE², LARRY BRADLEY³, MARINA BRADLEY⁴, KEANG-HAN HEUNG⁵, VICTORIA STRICK⁶, PAUL CHOI⁷, RACHA PETERSON-MANLEY⁸, ANA ZITLON⁹, ANA AVRAMOVA¹⁰, NARAYANA CHANDRA¹¹, SHIROHITO KAWAGUCHI¹², MASAYUKI OHTA¹³, GABRIEL B. BRANDES¹⁴, KEVIN SHARON¹⁵, MICHELLE TRINITY¹⁶, ROBERTO J. AVILA¹⁷, SARA OGAJI¹⁸, FELIX ANDRÁS-SANTOS¹⁹, DANIELA CARBONARA²⁰, CATHERINE CHRYST²¹, WILLIAM DAWSON²², BARBARA L. FOPP²³, ACHINT HOANG²⁴, CHRISTINE JOHAN²⁵, RAMON MARINER²⁶, MAMOH OUCHE²⁷, STEVEN A. RODNEY²⁸, DANIEL STARR²⁹, KAIICHI UEMURA³⁰

¹Space Telescope Science Institute, Baltimore, MD, USA
²Department of Physics, University of California, Davis, CA, 95616, USA
³Geneva Observatory, University of Geneva, CH, 5000 Maillette 51, 1205 Versoix, Switzerland
⁴Department of Astronomy, University of Michigan, 100 South University Ave., Ann Arbor, MI 48109, USA
⁵Physics Department, Ben-Gurion University of the Negev, P.O. Box 653, Beer-Sheva 84105, Israel
⁶Institute for Cosmic and Earth-Space Science, The University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa, Chiba 277-8581, Japan
⁷Department of Physics, Graduate School of Science, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
⁸Kavli Institute for the Physics and Mathematics at the University of Tsukuba, 1-1-1, Tsukuba, Ibaraki 305-8572, Japan
⁹Research Center for the Early Universe, The University of Tokyo, 5-1-5 Kashiwanoha, Tokyo 113-0033, Japan
¹⁰School of Physics, University of Melbourne, VIC 3010, Australia
¹¹Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA
¹²Lawrence Livermore National Laboratory, P.O. Box 808 L-210, Livermore, CA, 94551, USA
¹³Department of Astronomy, Steward Observatory, University of Arizona, 951 North Cherry Avenue, Flagstaff, AZ, 86021, USA
¹⁴Department of Physics and Astronomy, University of South Carolina, 712 Main St., Columbia, SC 29008, USA
¹⁵Institute of Astronomy and Astrophysics, Academia Sinica, PO Box 23141, Taipei 10617, Taiwan

Submitted to *ApJL* on 1/9/17

ABSTRACT

The most distant galaxies known are at $z \sim 10 - 11$, observed 400 - 500 Myr after the Big Bang. The few $z > 10$ candidates discovered to date have been exceptionally small, barely resolved, if at all, by the *Hubble Space Telescope*. Here we present the discovery of SPT0615-JD, a fortuitous $z \sim 10$ ($z_{Ly\alpha} = 9.9 \pm 0.6$) galaxy candidate stretched into an arc over $\sim 2.5''$ by the effects of strong gravitational lensing. Discovered in the Reionization Lensing Cluster Survey (RELICS) *Hubble* Treasury program and companion S-RELICS *Spitzer* program, this candidate has a lensed *H*-band magnitude of 23.7 ± 0.1 AB mag. With a magnification of $\mu \sim 4 - 7$ estimated from our lens models, the de-lensed intrinsic magnitude is 27.6 ± 0.3 AB mag, and the half-light radius is $r_{\text{e}} < 0.8$ kpc, both consistent with other $z > 9$ candidates. The inferred stellar mass ($\log(M_{\text{e}}/M_{\odot}) = 9.7^{+0.1}_{-0.2}$) and star formation rate ($\log(\text{SFR}_{\text{UV}}, \text{yr}^{-1}) = 1.3^{+0.2}_{-0.1}$) indicate that this candidate is a typical star-forming galaxy on the $z > 6$ SFR- M_{e} relation. We note that three independent lens models predict two counterimages, at least one of which should be of a similar magnitude to the arc, but these counterimages are not yet detected. Counterimages would not be expected if the arc were at lower redshift. However, the only spectral energy distributions capable of fitting the *Hubble* and *Spitzer* photometry well at lower redshifts require implausible combinations of $z \sim 2$ galaxy properties. The unprecedented lensed size of this $z \sim 10$ candidate offers the potential for the *James Webb Space Telescope* to study the geometric and kinematic properties of a galaxy observed 500 Myr after the Big Bang.

1. INTRODUCTION

With its high resolution and sensitivity, observations using the *Hubble Space Telescope* (HST) have sharpened our understanding of the high- z universe. Deep and wide extragalactic imaging surveys with ACS and WFC3 have uncovered thousands of galaxies at $z \geq 6$ in blank fields (see Finkelstein 2016; Stark 2016, for reviews), including the most distant galaxy found to date at $z = 11.1$ (GN-z11; Oesch et al. 2016). In addition, we have prioritized HST to observe the most massive galaxy clusters, taking advantage of the natural telescopes they create via strong gravitational lensing (CLASS; PI Postman; Frontier Fields; PI Lotz; RELICS; PI Coe). This investment in lensing fields has proven fruitful. We have discovered highly magnified (MACS1149-JD; Zheng et al. 2012; Hoag et al. 2017; MACS1145-JD and MACS1742-JD; Bowen et al. 2014; MACS0416-JD; Infante et al. 2015) and multiply-imaged galaxies (MACS0647-JD; Coe et al. 2013; A2744-JD; Zitrin et al. 2013) at redshifts up to $z \sim 10.8$, which have allowed us to study faint UV metal lines (Oesch et al. 2014; Riggs et al. 2015; Maseda et al. 2017), nebular emission lines (Smith et al. 2017; Stark et al. 2015; Hoag et al. 2017; Laporte et al. 2017), and the star formation rate density deep into the epoch of reionization (Oesch et al. 2014, 2017).

However, little is known in detail about the $z > 9$ universe, and the handful of candidates found so far exhibit peculiar properties. At $z \sim 11$, MACS0647-JD has a radius smaller than 100 pc, the size of the Giant Molecular Clouds in the local universe. GN-z11 is three times brighter than the characteristic UV luminosity (L_{UV}) of galaxies at that distance, a surprisingly bright given the CANDELS search area. Both $z \sim 10$ candidates MACS1145-JD and M0416-JD appear to have

1 bethm@stsci.edu



The high-level strategy remains the same...

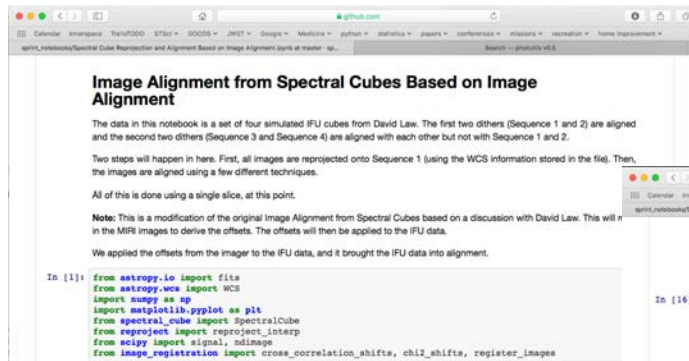
- Modular, open-source packages in python
- Community collaboration in development via Astropy and JDADF
- Scriptable libraries
- Jupyter notebooks with worked examples and embedded visualization
- Visualization tools



Jupyter Notebooks

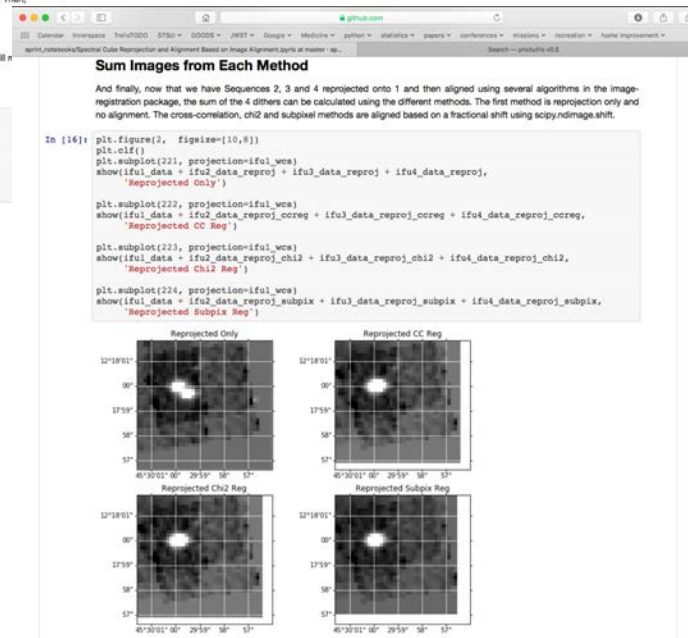
Browser based.

- Can be rendered online as static tutorials
- Users can download & customize



Renders text, equations, code & graphics
Built-in widget capability

Can be automatically tested to ensure consistency with code dependencies





But we are changing the underlying visualization technology...

- Fueled by data science, there has been rapid development of interactive visualization libraries
 - Python \Leftrightarrow JavaScript \Leftrightarrow Jupyter platform
- Many astronomy projects are creating JupyterLab environments and Jupyter Hubs to support users
 - LSST, NOAO, SciServer/HESARCH, PanGEO, ESAC/CADC, IPAC, DKIST, MAST,...
- **Technology is uniform throughout:**
 - There are advantages to having visualization technology that can be embedded in notebooks, in web browsers, or in desktop tools
- **This strategy enables re-use:**
 - Enhanced MAST tools can be based on the same architecture as the data-analysis tools
 - Widgets and GUIs can be re-used in notebooks, desktop apps, JupyterLab and archive
- **This is a forward-looking strategy, but consequently has more risk**
 - Proposed development strategy mitigates the risks



Top 5 Questions addressed in the past quarter

	Question	Answer	
1	Is the visualization performance acceptable, for JWST-sized data sets?	Yes	Qualitatively speaking
2	How much front-end development will we need to invest in to reach parity with the desktop (Qt-based) tools? a. Will we need to write a lot of javascript (or typescript)?	Some Not much	More than for Qt but not scary
3	Are the core dependencies (libraries & frameworks) mature enough? a. Are they likely to survive & grow over the next few years? b. Will it be hard to add new functionality if needed?	Almost Some No	Details on later slides...
4	Can we get the notebook workflow and App/quicklook workflow to work together well enough that it makes sense to build them together?	Yes	Biggest success
5	How easy and reliable is it to configure the UI layout/styling/colors/fonts?	Moderately	Details on later slides...



Demos (possibilities)

- Cubeviz prototype in a notebook with a little bit of science workflow
- Cubeviz prototype in a web browser
- Cubeviz prototype as a desktop App
- Cubeviz prototype as a MAST App



Development strategy affected on a subset of the packages

Package	Purpose	Maturity Now	Maturity 2020
Astropy	A community python library for astronomy		
glue	Multidimensional linked-dataset exploration; visualization toolkit		
ginga	Image viewer toolkit		TBD
Photutils	Source detection and photometry		
astroimtools	Convenience tools for working with astronomical images		
imexam	Simple image examination and plotting, similar functionality to IRAF's imexamine		
specutils	Spectroscopic utilities and data containers		
SpecViz	Interactive 1D spectral analysis tool with similar functionality to IRAF's splot		
MOSViz	Quick-look analysis for multi-object spectroscopy		
CubeViz & spectral_cube	3D spectroscopic analysis		
ASDF	Advanced Scientific Data Format - next-generation interchange format for scientific data (FITS replacement)		
gwcs	Generalized world coordinate system tools for dealing with image and spectral geometries		
synphot	Synthetic photometry toolkit for building model spectra and estimating count rates		

Ginga questions are:

- *Where does it stand in priority for resources?*
- *What do Astronomers want for the UI?*
- *Will they demand ds9?*

Maturity

These take a step backwards, although we will re-use a lot of existing 'viz code.

We could probably get them up to parity with the Qt tools **and working for cloud+desktop+notebook+MAST applications** in 6 months if that were the main focus, but we argue for a different balance of resources, which will stretch this out.



Different layers in the Data Analysis Software Stack

Notebooks collection – Jupyter notebooks illustrating a particular workflow. 36 identified so far.

Viz Application collection – GUI based tools that link visualization and analysis. Work in browsers, as a desktop application, embedded as windows in JupyterLab, embedded as cells in a Jupyter notebook, or embedded on a MAST website.

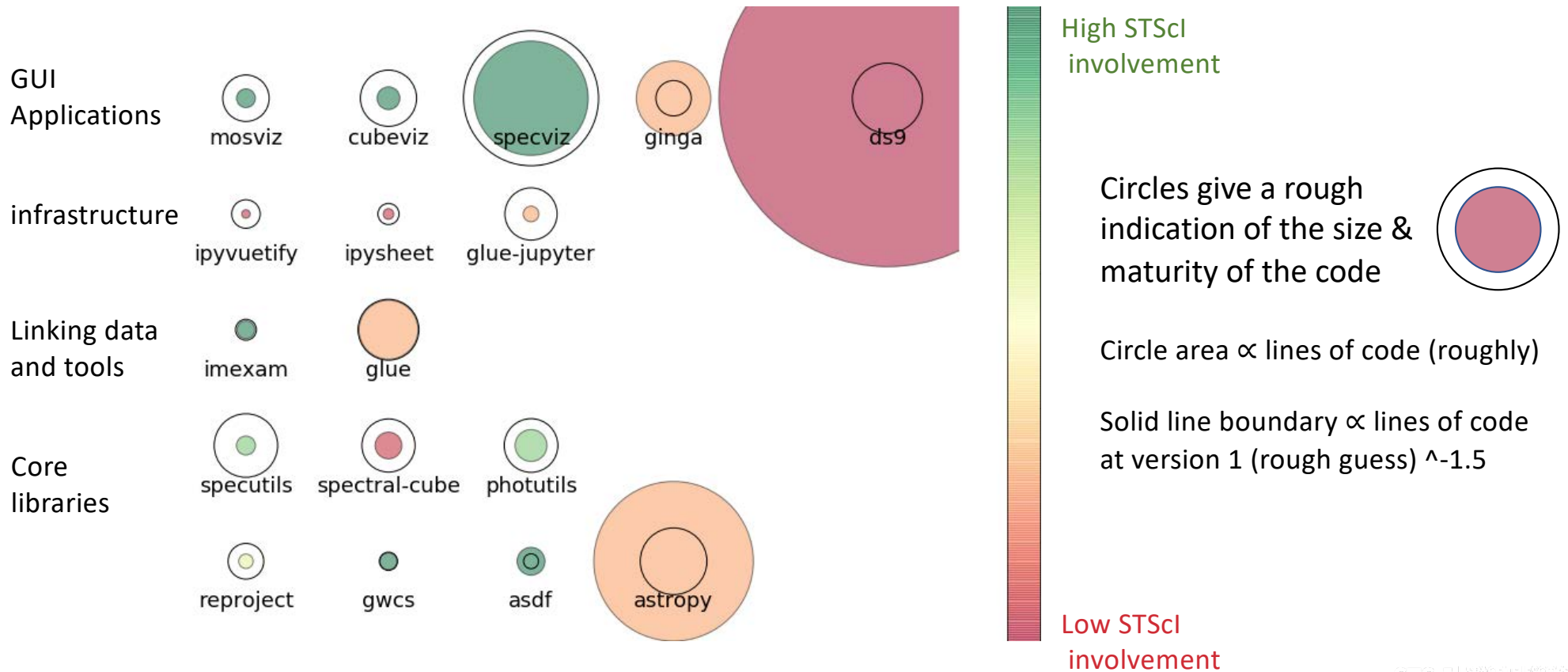
Widget collection - Modular building blocks of the Viz libraries. They typically have no menu, or only a very simple menu.

Core Viz libraries – Support the widgets and visualization tools.

Core numerical and astronomy libraries – Responsible for numerical computation and i/o. All the real scientific calculations happen here. All I/O to and from disk passes through these libraries to ensure consistency of data formats.



Visualization of current status the software stack





Development strategy

Principles:

- Agile approach – iterative development with close involvement of astronomers (mostly from INS)
- Get useful tools to the community fast and incrementally, even if the functionality in each increment is fairly limited.
- Get feedback on the tools
- Use JWST workflows to drive development
- Multiple workflows will be developed in parallel. We will thus have to phase releases of the underlying libraries to support the set of notebooks planned for the next releases.



Notebooks-first approach

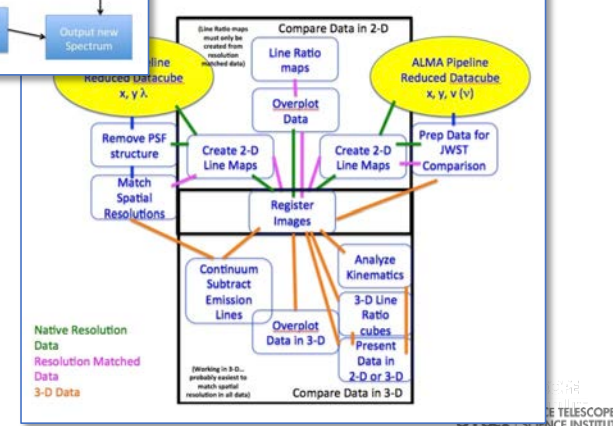
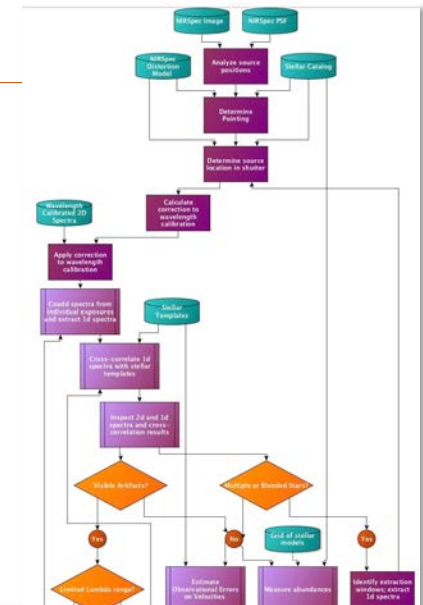
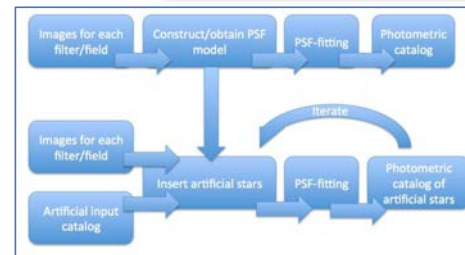
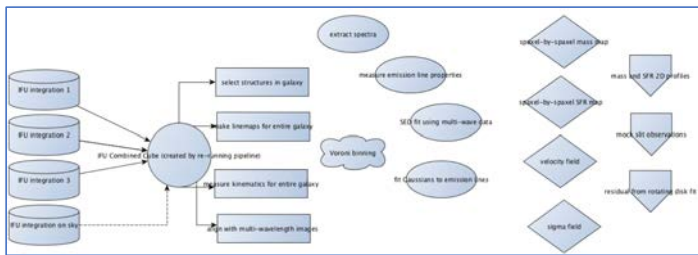
- Sketch out a JWST workflow in a notebook
 - which may or may not work or be pretty in the first iteration
 - Test & debate algorithms based on working notebooks.
 - If visualizations are needed at this stage, make them simple and probably static. Or sketches/descriptions of desired visualization & interactive functionality.
- Implement the algorithm in the core library
 - This strategy encourages developing user-friendly APIs early
- Redo the notebook calling the library
- Augment the notebook with interactive visualizations where sensible
- Guided by this, prioritize viz development:
 - New widget visualization capabilities
 - Exposing additional analysis tasks via menus
 - Work on core viz-libraries to enable something we haven't already thought of



JWST workflows

Descriptions of post-pipeline analysis for different science use cases. Most written in 2014, by mix of STScI, GTO, and other Astronomers. More welcome.

- Consist of
 - Narrative
 - List of inputs
 - Flow diagram
 - Identification of existing tools
 - Wishlist of tools or features



<https://outerspace.stsci.edu/display/JWSTDATF/Data+Analysis+Tools+Use+Cases>



Documentation Strategy

JDox

- A lightweight entry point for JWST users pointing to the documentation

Readthedocs.io hosts the documentation

- Built and maintained in the same repository as the code
- Examples in documentation can be tested as the documents are built
 - ▶ Helps uncover inconsistencies between documentation & code
- Built using sphinx, which is the de-facto standard for Python
 - ▶ Familiar to python users
 - ▶ Better rendering of code than Jdox
- Styling still evolving

Jupyter notebooks

- Open-source web application for documents that contain live code, equations, visualizations and narrative text.
- Excellent for tutorials
 - Documentation for complex workflows is interleaved with the actual executable steps



Readthedocs

The screenshot shows the specutils.readthedocs.io website. The page title is "Working with Spectrum1Ds". It features a "Page Contents" sidebar on the left with links to "Basic Spectrum Creation", "Reading from a File", "Including Uncertainties", "Defining WCS", "Providing a FITS-style WCS", "Multi-dimensional Data Sets", "Reference/API", "Classes", and "Class Inheritance Diagram". The main content area is titled "Working with Spectrum1Ds" and includes a section for "Basic Spectrum Creation". This section contains a code block with Python code for creating a Spectrum1D object and plotting it. Below the code is a line plot of Flux vs. Dispersion. At the bottom of the page, there is a section titled "Reading from a File" with a "View source" button.

specutils.readthedocs.io

Searchable

Integrated text, code, graphics
Code tested as docs are built

Downloadable source code

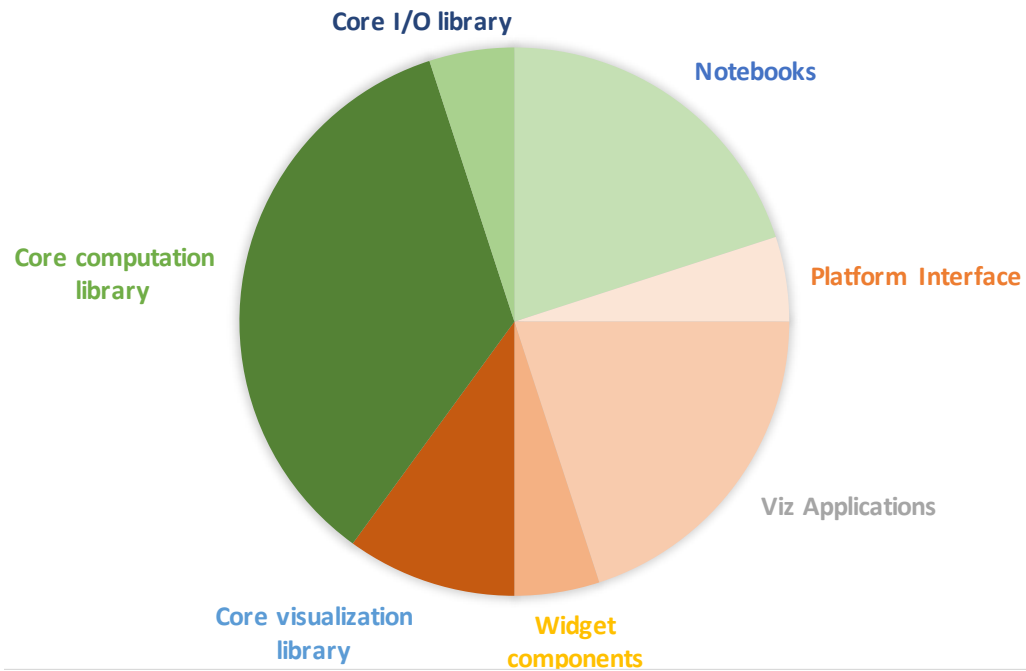
Documentation automatically
tagged to code versions

API documentation pulled from
source code docstrings (helps
ensure consistency)



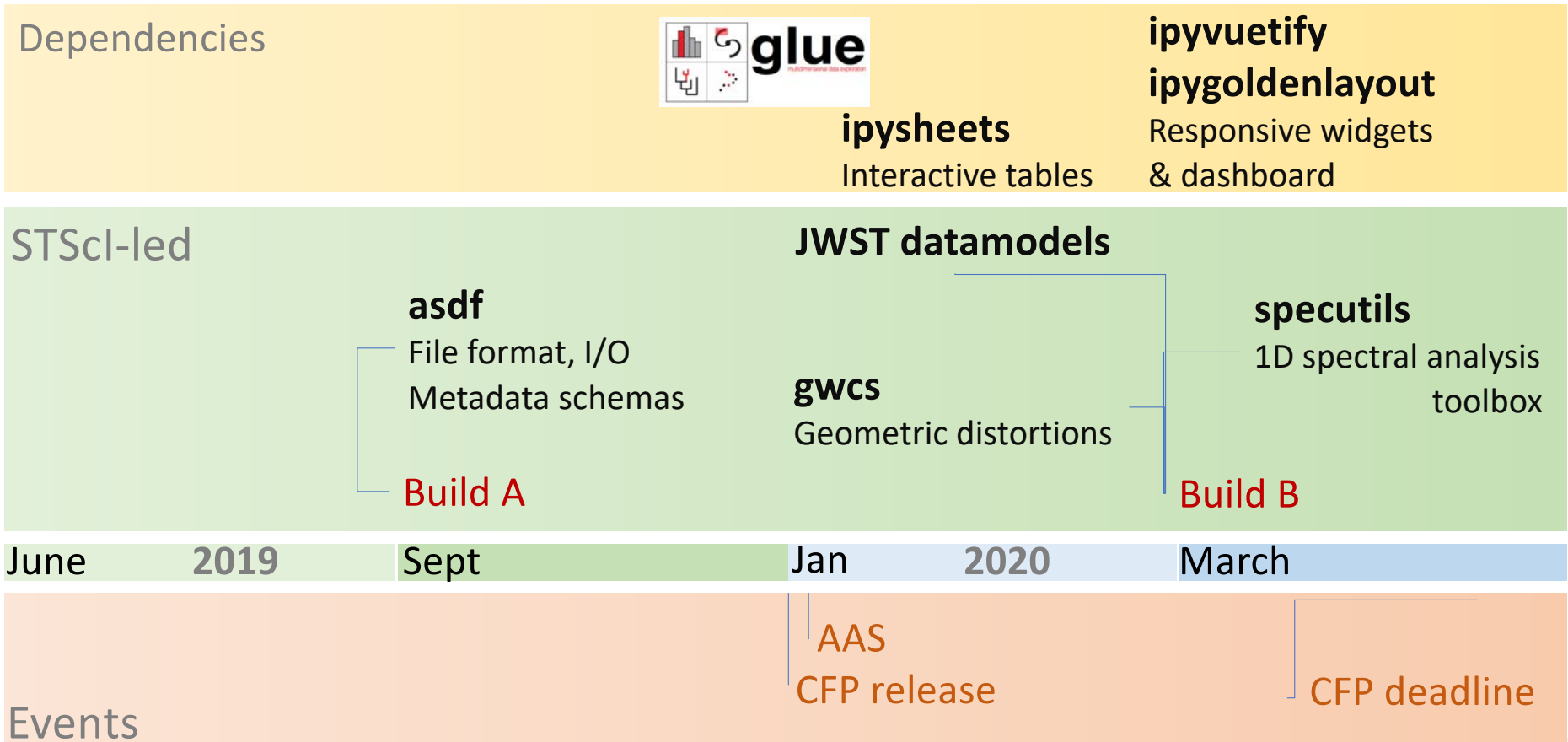
Resource allocations

DEVELOPMENT RESOURCE FRACTIONS





Timeline





Timeline

Dependencies

voila

App builder

spectral-cube

3D spectral analysis toolbox

STScI-led

imexam

Interactive
image analysis

photutils

detection
photometry

specviz

Interactive 1D spectral analysis

mosviz

MOS quicklook

reproject

Rectification & combination

cubeviz

Interactive 3D spectral analysis

notebooks

Data Analysis tutorials

imviz

Image display in notebook

Build C

Build D

June 2020

Sept

Jan 2021

March

AAS

AAS

Launch

Training workshops

Virtual training

Events



Offramp for non-Python users

- File format
 - JWST files from MAST are FITS with ASDF is just in an extension
 - Document the headers & file formats (in progress)
 - Facilitate using ASDF (in progress)
 - Better schema documentation
 - ASDF.info() and similar tools for inspecting & editing headers
 - ASDF image viewer (starting with ds9 & ginga)
 - Work with community to build reader/writers in other languages (C++ started)
 - Work with community to build reader/writer for IDL or interface python to it (no effort yet)
- ds9 support
 - Stage 1 image loading via XPA, with helper for browsing file (started experimenting)
 - Stage 2: Add polynomial approximation to FITS WCS to improve ds9 distortion handling
 - Stage 3: Embedded python asdf support in ds9
 - Stage 4: ASDF read-only support in C





“I don’t like python...give me simple tools like IRAF/pyraf”

Some of this stems from impatience with boilerplate:

- Import statements
- File I/O
- `if __name__=="__main__":`

Strategies to satisfy this user base

- Training
- Pre-configured imports (reduces common imports to one line)
- Custom configs of shell environments (ipython/jupyter, [xonsh](#) or others)
- File-based workflows
 - Helper packages that roll together common workflows
 - Simple wrappers around memory-oriented tools to do the I/O before & after
- Unix command-line file-based interfaces
 - using [fire](#), [begins](#), [docopt](#), or similar libraries
- Chain tasks into common higher-level workflows

This all comes downstream from building the python modules, functions & classes



Image displays

Desktop tools:

- ds9 will likely still be the most popular for the first few cycles
 - Support through stage 2 looks feasible in time for cycle 1
- Ginga is more easily extensible and will have better support for gWCS & ASDF
 - Main use case is inspecting sources in individual dithers
 - Ds9 with SIP coefficients is probably good enough for this use case

Jupyter-compatible tools:

- Basic ginga already works in the browser (& Jupyter notebooks & with glue)
- Other options exist (e.g. bqplot image viewer)
 - Add functionality – especially for blinking & locking images



Risks

High impact, low probability	High impact, high probability
Glue development/maintenance stops	Missing simulated data in pipeline output formats (can't test all use cases)
Community abandons Python or Jupyter for the next shiny thing	Loss of expert developers (mitigate by cross-training)
Low impact, low probability	Low impact, high probability
Community develops tools with same functionality (great...but try to avoid surprises here)	A software stack item becomes a dead end (mitigate by abstracting interfaces where practical)
	FTE shortfalls (prioritize)



Contingency

- Qt visualization tools aren't going anywhere
 - Specviz, MOSViz, & Cubeviz
 - They are currently available to the community to use
 - Via astroconda
 - Providing user support via helpdesk, github, etc
 - Core functionality goes into the underlying non-viz libraries regardless of visualization technology
- We can revive development of the Qt tools without much problem a year hence if needed.



Takeaway messages

- Community input on JWST data-analysis tools is welcome at all levels from prioritization to open-source code contributions
- Jupyter notebooks will play a big role in our data-analysis support
- We are changing visualization technology to enhance re-use between desktop, notebooks, and remote workflows