The struggles of

Developing Python Packages for Spectroscopy

A small overview of the lessons learned in developing open-source spectroscopy packages.

Nicholas Earl
Space Telescope Science Institute
PyAstro 2018

STScI | SPACE TELESCOPE SCIENCE INSTITUTE specutils

An astropy package for spectroscopy

Analysis & Data Handling

SpecViz

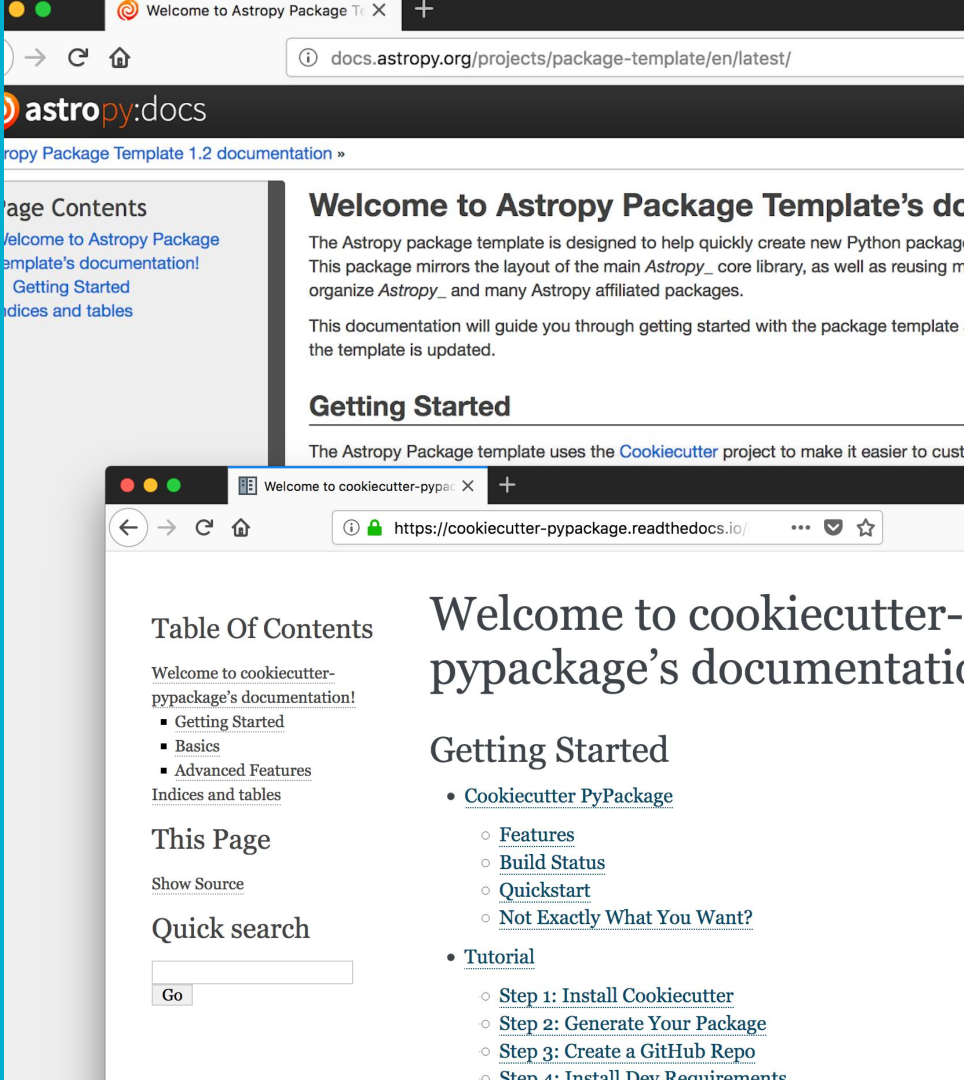
Visualization & Interaction

Cosmoscope

Multi-source distributed communication

Lesson 1

The easier a package is to get into the hands of users, the easier it is to get constructive feedback on



Welcome to Astropy Package Te

docs.astropy.org/projects/package-template/en/latest/

astropy:docs

ropy Package Template 1.2 documentation »

'age Contents

/elcome to Astropy Package emplate's documentation!

Getting Started

ndices and tables

Welcome to Astropy Package Template's do

The Astropy package template is designed to help quickly create new Python packag This package mirrors the layout of the main *Astropy_* core library, as well as reusing m organize *Astropy_* and many Astropy affiliated packages.

This documentation will guide you through getting started with the package template the template is updated.

Getting Started

The Astropy Package template uses the Cookiecutter project to make it easier to cust

Welcome to cookiecutter-pypac

https://cookiecutter-pypackage.readthedocs.io/

Table Of Contents

Welcome to cookiecutter-pypackage's documentation!

- Getting Started
- Basics
- Advanced Features

Indices and tables

This Page

Show Source

Quick search

Go

Welcome to cookiecutter-pypackage's documentatio

Getting Started

- Cookiecutter PyPackage
  - Features
  - Build Status
  - Quickstart
  - Not Exactly What You Want?
- Tutorial
  - Step 1: Install Cookiecutter
  - Step 2: Generate Your Package
  - Step 3: Create a GitHub Repo
  - Step 4: Install Dev Requirements

Distribution platforms

ANACONDA®

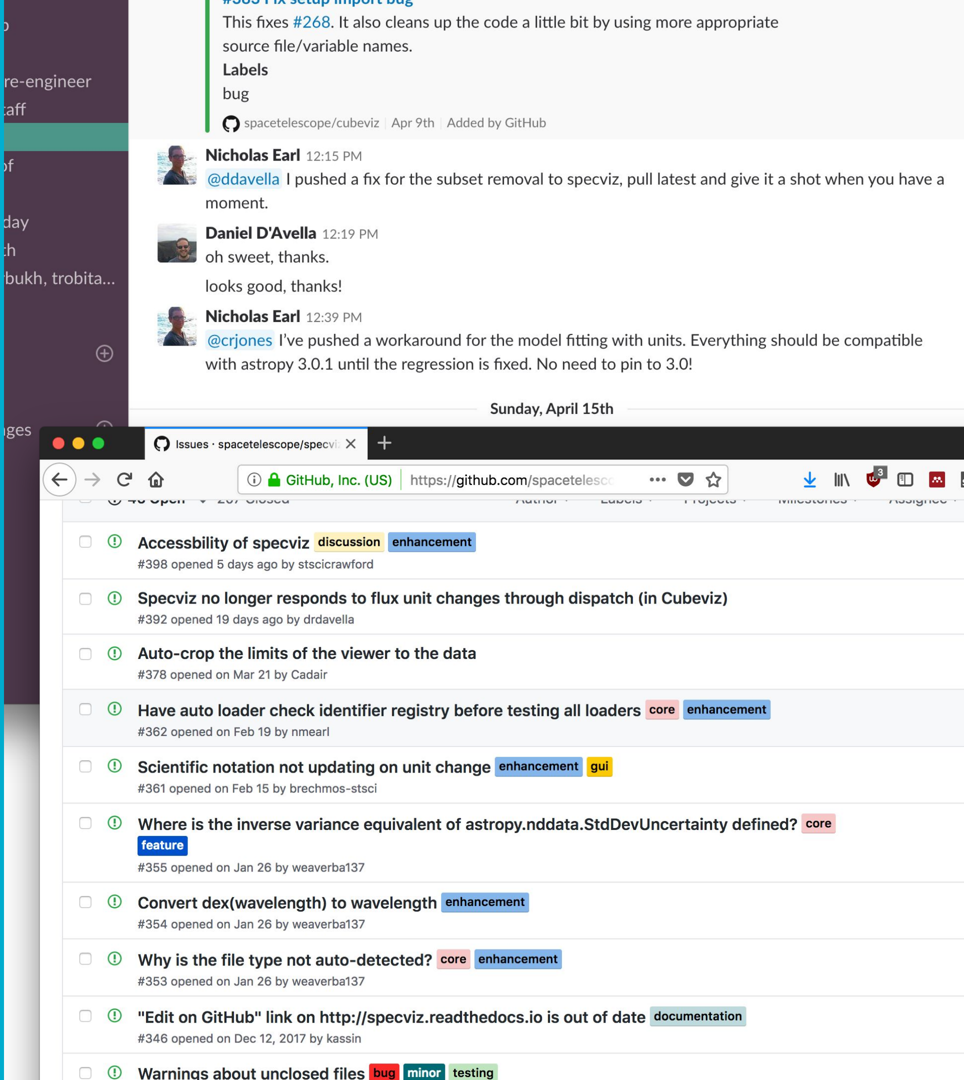
- Language-agnostic packaging tool and installer
- Environment manager
- Requires extra legwork

python™ Package Index

- Python-package centric
- Installs from source (2.x+) or from binaries (3.x+)
- Built-in

Lesson 2

A voice spoken is a voice heard, and other methods of unadulterated feedback



This fixes #268. It also cleans up the code a little bit by using more appropriate source file/variable names.

Labels

bug

spacetelescope/cubeviz | Apr 9th | Added by GitHub

Nicholas Earl 12:15 PM
@ddavella I pushed a fix for the subset removal to specviz, pull latest and give it a shot when you have a moment.

Daniel D'Avella 12:19 PM
oh sweet, thanks.

looks good, thanks!

Nicholas Earl 12:39 PM
@crjones I've pushed a workaround for the model fitting with units. Everything should be compatible with astropy 3.0.1 until the regression is fixed. No need to pin to 3.0!

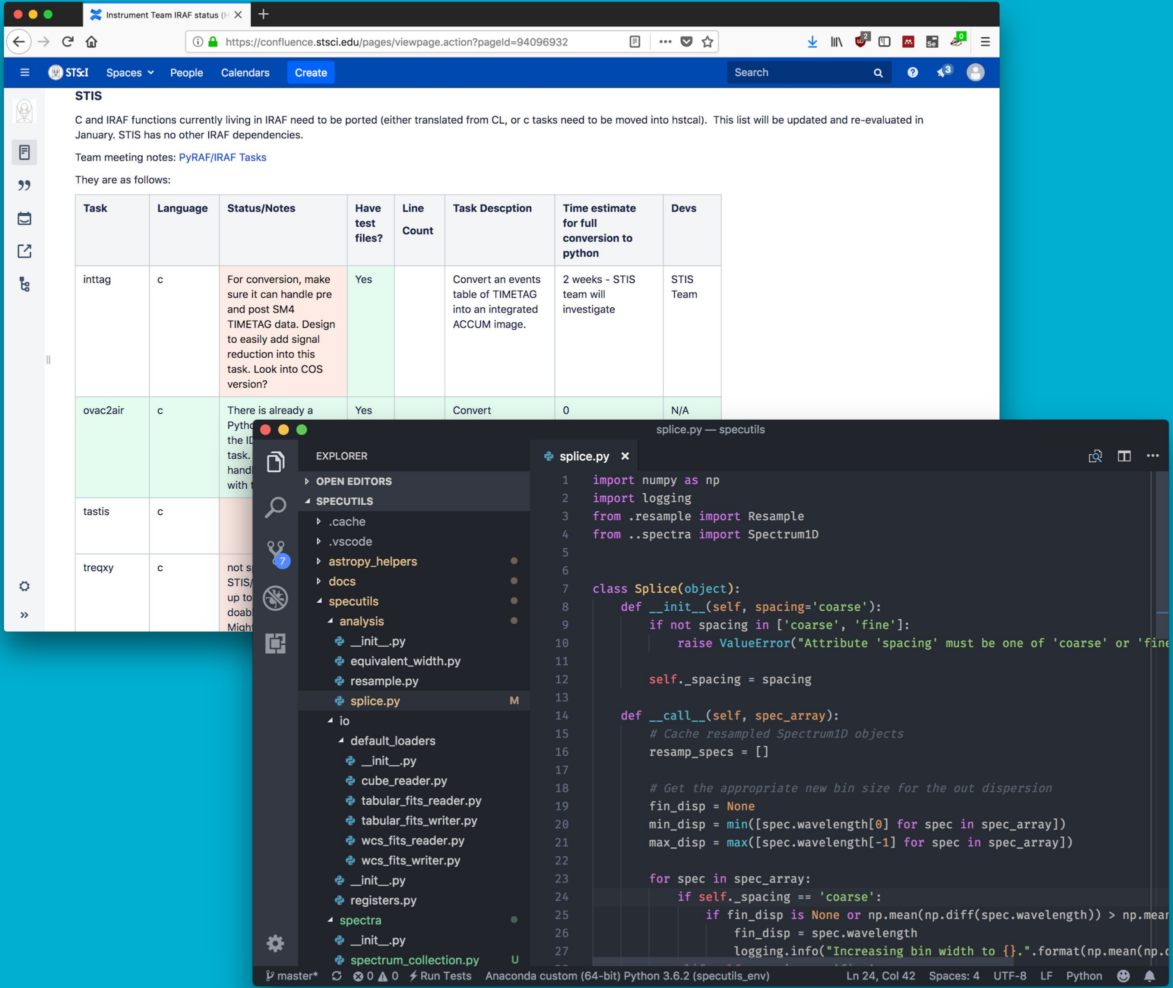
Sunday, April 15th

Issues · spacetelescope/specvi
GitHub, Inc. (US)
https://github.com/spacetelesc

- □ Accessbility of specviz `discussion` `enhancement`
  #398 opened 5 days ago by stscicrawford
- □ Specviz no longer responds to flux unit changes through dispatch (in Cubeviz)
  #392 opened 19 days ago by drdavella
- □ Auto-crop the limits of the viewer to the data
  #378 opened on Mar 21 by Cadair
- □ Have auto loader check identifier registry before testing all loaders `core` `enhancement`
  #362 opened on Feb 19 by nmearl
- □ Scientific notation not updating on unit change `enhancement` `gui`
  #361 opened on Feb 15 by brechmos-stsci
- □ Where is the inverse variance equivalent of astropy.nddata.StdDevUncertainty defined? `core` `feature`
  #355 opened on Jan 26 by weaverba137
- □ Convert dex(wavelength) to wavelength `enhancement`
  #354 opened on Jan 26 by weaverba137
- □ Why is the file type not auto-detected? `core` `enhancement`
  #353 opened on Jan 26 by weaverba137
- □ "Edit on GitHub" link on http://specviz.readthedocs.io is out of date `documentation`
  #346 opened on Dec 12, 2017 by kassin
- □ Warnings about unclosed files `bug` `minor` `testing`

Hands-on Feature Sprints & Hack Days

- Individuals with vested interest that may not have development skills
- Maintainers of previous iterations of features that may not be written in python
- Teams that have specific workflows that'd like replicated



Instrument Team IRAF status
https://confluence.stsci.edu/pages/viewpage.action?pageId=94096932
STScI
Spaces
People
Calendars
Create
Search

STIS

C and IRAF functions currently living in IRAF need to be ported (either translated from CL, or c tasks need to be moved into hstcal). This list will be updated and re-evaluated in January. STIS has no other IRAF dependencies.

Team meeting notes: PyRAF/IRAF Tasks

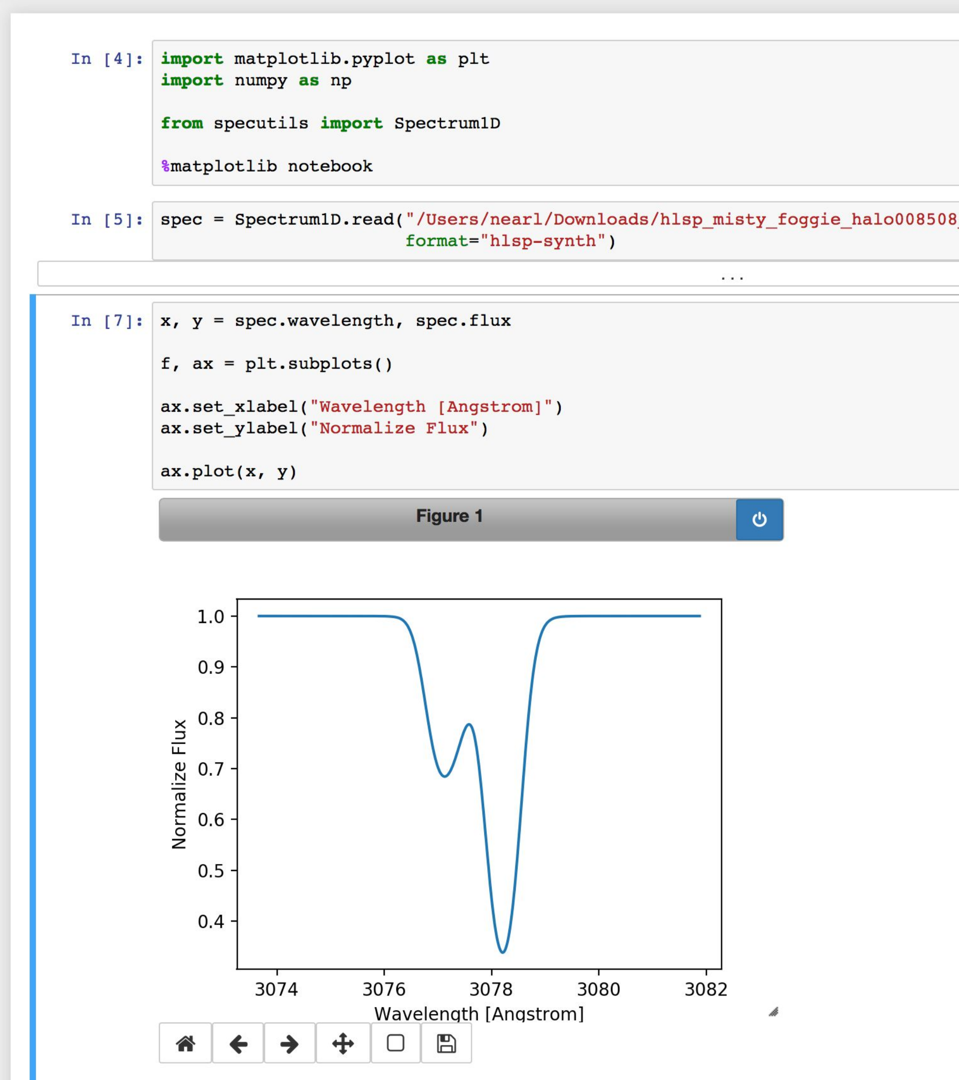
They are as follows:

Task	Language	Status/Notes	Have test files?	Line Count	Task Descption	Time estimate for full conversion to python	Devs
inttag	c	For conversion, make sure it can handle pre and post SM4 TIMETAG data. Design to easily add signal reduction into this task. Look into COS version?	Yes		Convert an events table of TIMETAG into an integrated ACCUM image.	2 weeks - STIS team will investigate	STIS Team
ovac2air	c	There is already a Pytho the ID task. hand with t	Yes		Convert	0	N/A
tastis	c						
treqxy	c	not s STIS/ up to doab Migh					

```python
import numpy as np
import logging
from .resample import Resample
from ..spectra import Spectrum1D


class Splice(object):
    def __init__(self, spacing='coarse'):
        if not spacing in ['coarse', 'fine']:
            raise ValueError("Attribute 'spacing' must be one of 'coarse' or 'fine

        self._spacing = spacing

    def __call__(self, spec_array):
        # Cache resampled Spectrum1D objects
        resamp_specs = []

        # Get the appropriate new bin size for the out dispersion
        fin_disp = None
        min_disp = min([spec.wavelength[0] for spec in spec_array])
        max_disp = max([spec.wavelength[-1] for spec in spec_array])

        for spec in spec_array:
            if self._spacing == 'coarse':
                if fin_disp is None or np.mean(np.diff(spec.wavelength)) > np.mear
                    fin_disp = spec.wavelength
                    logging.info("Increasing bin width to {}.".format(np.mean(np.c
```

Lesson 3

Respect users' workflows by ensuring piecemeal package adoption (because some usage is better than none)



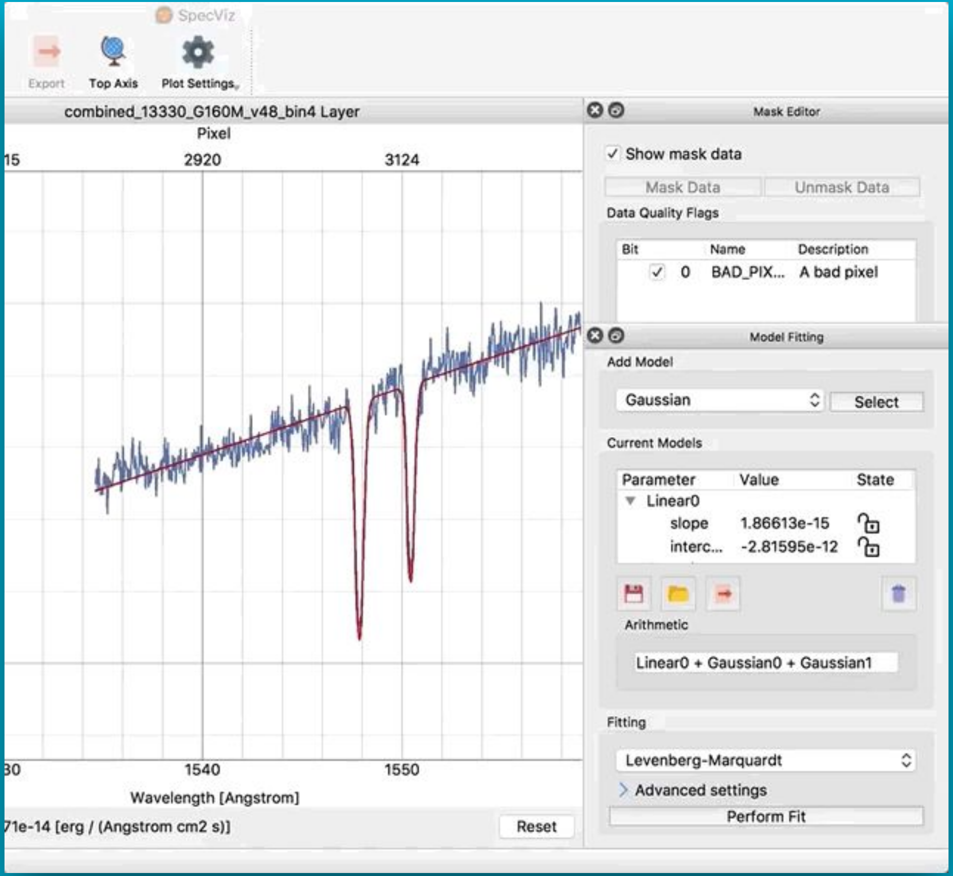
In [4]:
```
import matplotlib.pyplot as plt
import numpy as np

from specutils import Spectrum1D

%matplotlib notebook
```

In [5]:
```
spec = Spectrum1D.read("/Users/nearl/Downloads/hlsp_misty_foggie_halo008508
                       format="hlsp-synth")
```

...

In [7]:
```
x, y = spec.wavelength, spec.flux

f, ax = plt.subplots()

ax.set_xlabel("Wavelength [Angstrom]")
ax.set_ylabel("Normalize Flux")

ax.plot(x, y)
```

Avoiding a Walled-Garden



☑ Show mask data

Data Quality Flags

	Bit	Name	Description
✓	0	BAD_PIX...	A bad pixel

Add Model: Gaussian

Current Models

Parameter	Value	State
Linear0		
slope	1.86613e-15	
interc...	-2.81595e-12	

Arithmetic: Linear0 + Gaussian0 + Gaussian1

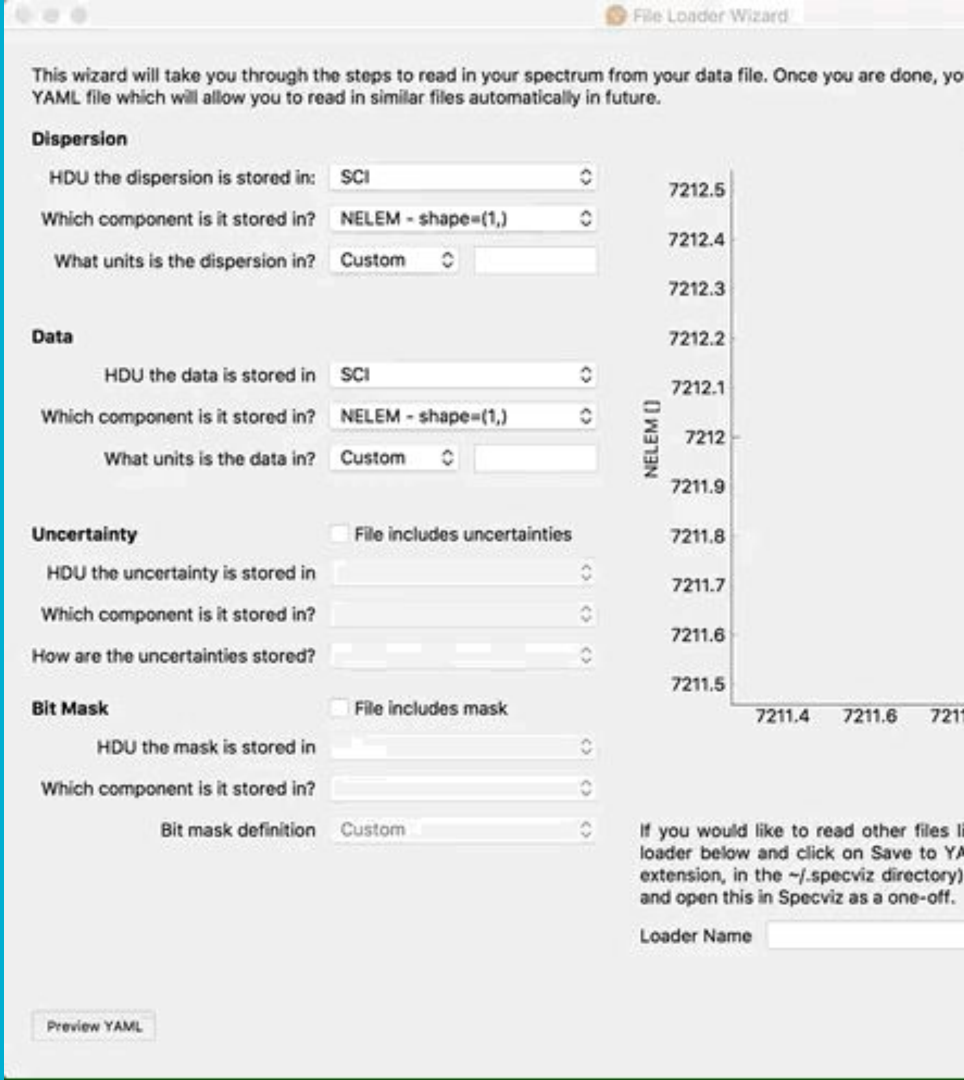
Fitting: Levenberg-Marquardt

Advanced settings

- Exportable, Astropy-ready data models usable outside SpecViz
- Easy, extensible IO infrastructure for parsing spectral data into Astropy quantities

Lesson 4

Give users the tools they need to implement features they want (so you don't have to)



File Loader Wizard

This wizard will take you through the steps to read in your spectrum from your data file. Once you are done, yo
YAML file which will allow you to read in similar files automatically in future.

Dispersion

HDU the dispersion is stored in: SCI

Which component is it stored in? NELEM - shape=(1,)

What units is the dispersion in? Custom

Data

HDU the data is stored in SCI

Which component is it stored in? NELEM - shape=(1,)

What units is the data in? Custom

Uncertainty □ File includes uncertainties

HDU the uncertainty is stored in

Which component is it stored in?

How are the uncertainties stored?

Bit Mask □ File includes mask

HDU the mask is stored in

Which component is it stored in?

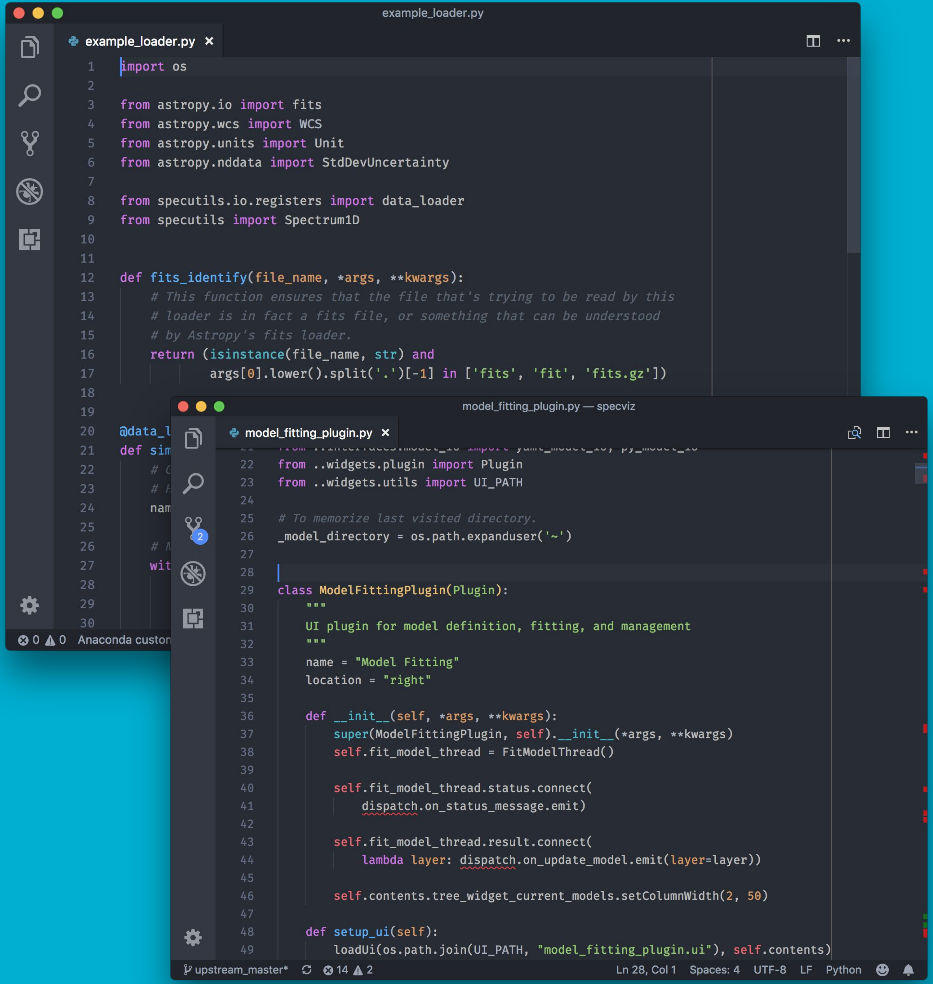
Bit mask definition Custom

If you would like to read other files li
loader below and click on Save to YA
extension, in the ~/.specviz directory)
and open this in Specviz as a one-off.

Loader Name

Help others to help you help them help themselves

- Where sensible, abstract behavior away from the package mechanics
- Compartmentalize features to make their creation simple to integrate later

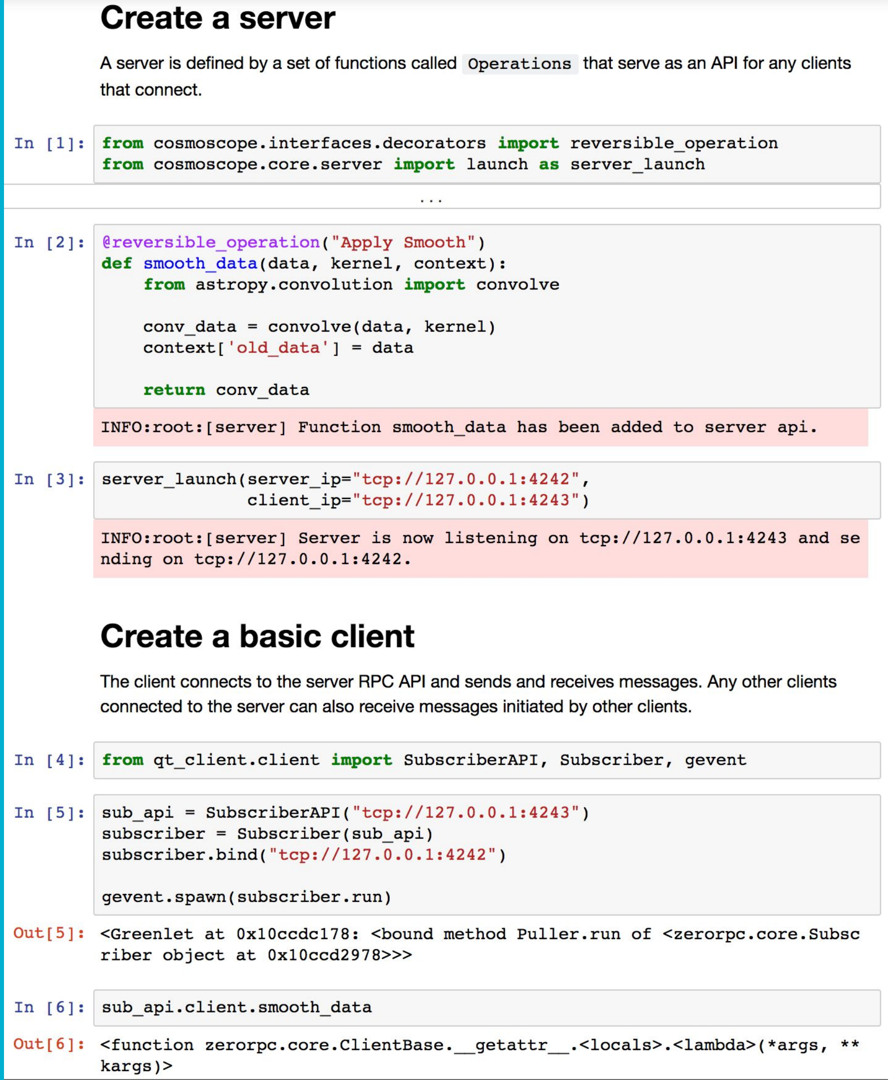


```python
import os

from astropy.io import fits
from astropy.wcs import WCS
from astropy.units import Unit
from astropy.nddata import StdDevUncertainty

from specutils.io.registers import data_loader
from specutils import Spectrum1D


def fits_identify(file_name, *args, **kwargs):
    # This function ensures that the file that's trying to be read by this
    # loader is in fact a fits file, or something that can be understood
    # by Astropy's fits loader.
    return (isinstance(file_name, str) and
            args[0].lower().split('.')[-1] in ['fits', 'fit', 'fits.gz'])


@data_l
def sim
    # G
    # H
    nam

    # M
    wit
```

```python
from ..interfaces.model_io import yaml_model_io, py_model_io
from ..widgets.plugin import Plugin
from ..widgets.utils import UI_PATH

# To memorize last visited directory.
_model_directory = os.path.expanduser('~')


class ModelFittingPlugin(Plugin):
    """
    UI plugin for model definition, fitting, and management
    """
    name = "Model Fitting"
    location = "right"

    def __init__(self, *args, **kwargs):
        super(ModelFittingPlugin, self).__init__(*args, **kwargs)
        self.fit_model_thread = FitModelThread()

        self.fit_model_thread.status.connect(
            dispatch.on_status_message.emit)

        self.fit_model_thread.result.connect(
            lambda layer: dispatch.on_update_model.emit(layer=layer))

        self.contents.tree_widget_current_models.setColumnWidth(2, 50)

    def setup_ui(self):
        loadUi(os.path.join(UI_PATH, "model_fitting_plugin.ui"), self.contents)
```

Lesson 5

Choice, choice, choice; expand users' ability to leverage the spectroscopic packages in multiple ways



Create a server

A server is defined by a set of functions called `Operations` that serve as an API for any clients that connect.

```
In [1]: from cosmoscope.interfaces.decorators import reversible_operation
        from cosmoscope.core.server import launch as server_launch
```

...

```
In [2]: @reversible_operation("Apply Smooth")
        def smooth_data(data, kernel, context):
            from astropy.convolution import convolve

            conv_data = convolve(data, kernel)
            context['old_data'] = data

            return conv_data

        INFO:root:[server] Function smooth_data has been added to server api.
```

```
In [3]: server_launch(server_ip="tcp://127.0.0.1:4242",
                      client_ip="tcp://127.0.0.1:4243")

        INFO:root:[server] Server is now listening on tcp://127.0.0.1:4243 and se
        nding on tcp://127.0.0.1:4242.
```

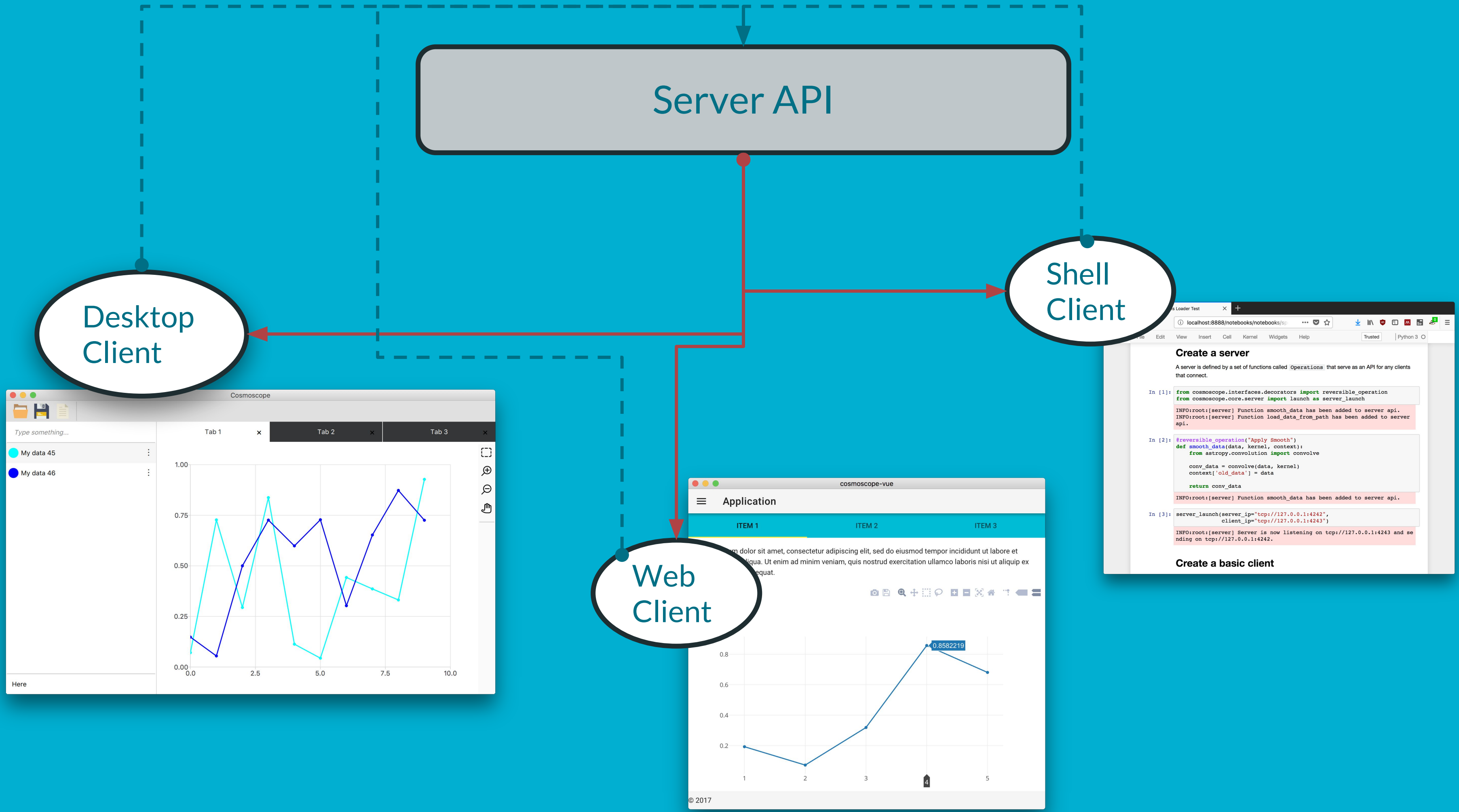
Create a basic client

The client connects to the server RPC API and sends and receives messages. Any other clients connected to the server can also receive messages initiated by other clients.

```
In [4]: from qt_client.client import SubscriberAPI, Subscriber, gevent
```

```
In [5]: sub_api = SubscriberAPI("tcp://127.0.0.1:4243")
        subscriber = Subscriber(sub_api)
        subscriber.bind("tcp://127.0.0.1:4242")

        gevent.spawn(subscriber.run)

Out[5]: <Greenlet at 0x10ccdc178: <bound method Puller.run of <zerorpc.core.Subsc
        riber object at 0x10ccd2978>>>
```

```
In [6]: sub_api.client.smooth_data

Out[6]: <function zerorpc.core.ClientBase.__getattr__.<locals>.<lambda>(*args, **
        kargs)>
```



Server API

Desktop Client

Shell Client

Web Client

…m dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et …liqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex …equat.

Create a server

A server is defined by a set of functions called `Operations` that serve as an API for any clients that connect.

```
In [1]: from cosmoscope.interfaces.decorators import reversible_operation
        from cosmoscope.core.server import launch as server_launch
```

```
INFO:root:[server] Function smooth_data has been added to server api.
INFO:root:[server] Function load_data_from_path has been added to server
api.
```

```
In [2]: @reversible_operation("Apply Smooth")
        def smooth_data(data, kernel, context):
            from astropy.convolution import convolve

            conv_data = convolve(data, kernel)
            context['old_data'] = data

            return conv_data
```

```
INFO:root:[server] Function smooth_data has been added to server api.
```

```
In [3]: server_launch(server_ip="tcp://127.0.0.1:4242",
                      client_ip="tcp://127.0.0.1:4243")
```

```
INFO:root:[server] Server is now listening on tcp://127.0.0.1:4243 and se
nding on tcp://127.0.0.1:4242.
```

Create a basic client

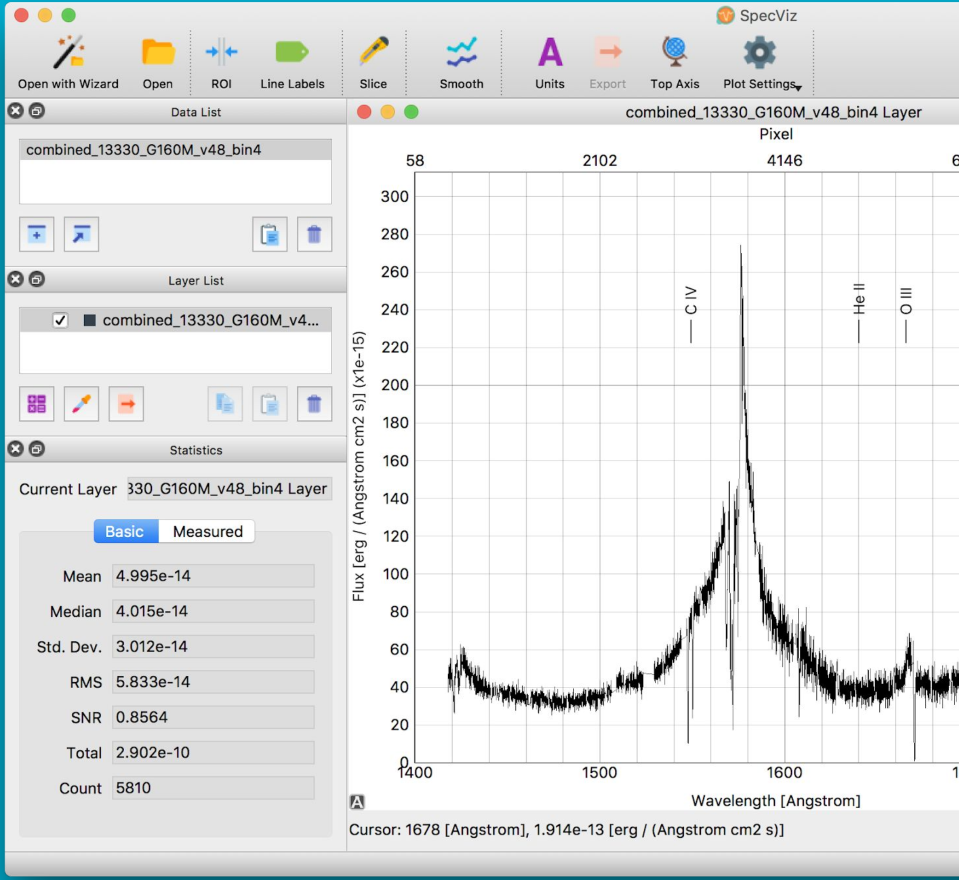
Questions?

SpecViz: /spacetelescope/specviz

- New release v0.5, now includes user-loadable line list labels that pin to plots.
- Performance improvements to loading data, bug fixes with arithmetic propagation.
- And more.

SpecUtils: /astropy/specutils

- New v0.3 release, will be available on PyPI soon!



Current Layer 330_G160M_v48_bin4 Layer

Mean	4.995e-14
Median	4.015e-14
Std. Dev.	3.012e-14
RMS	5.833e-14
SNR	0.8564
Total	2.902e-10
Count	5810

```
Cursor: 1678 [Angstrom], 1.914e-13 [erg / (Angstrom cm2 s)]
```