Widening the Spectral Processing Aperture with ENVI and Open-Source Tools

Austin Coates

Solutions Engineering Manager | L3Harris

Austin.Coates@l3Harris.com

SPECTRAL SESSIONS

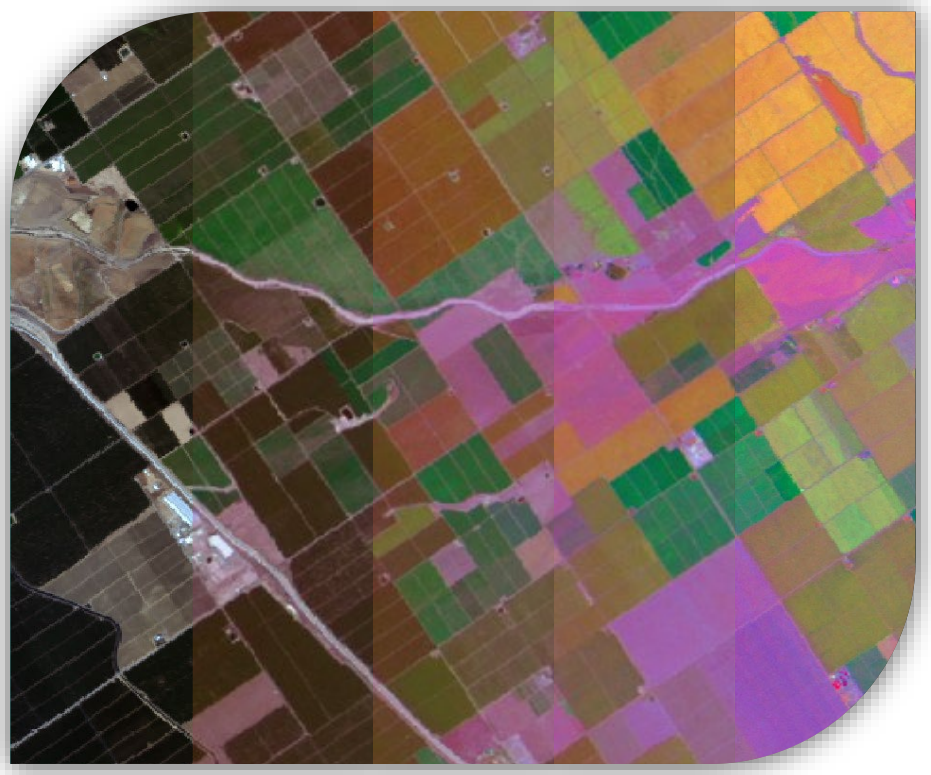
Speaker

Austin Coates
Solutions Engineering Manager
Austin.Coates@L3Harris.com

L3Harris Geospatial
L3HarrisGeospatial.com
geospatialinfo@l3harris.com
303-786-9900

Agenda

- Simplified Analysis Workflow
- Entry Points
- Example
- Questions



Farmland True Color To Principal Components

SPECTRAL SESSIONS

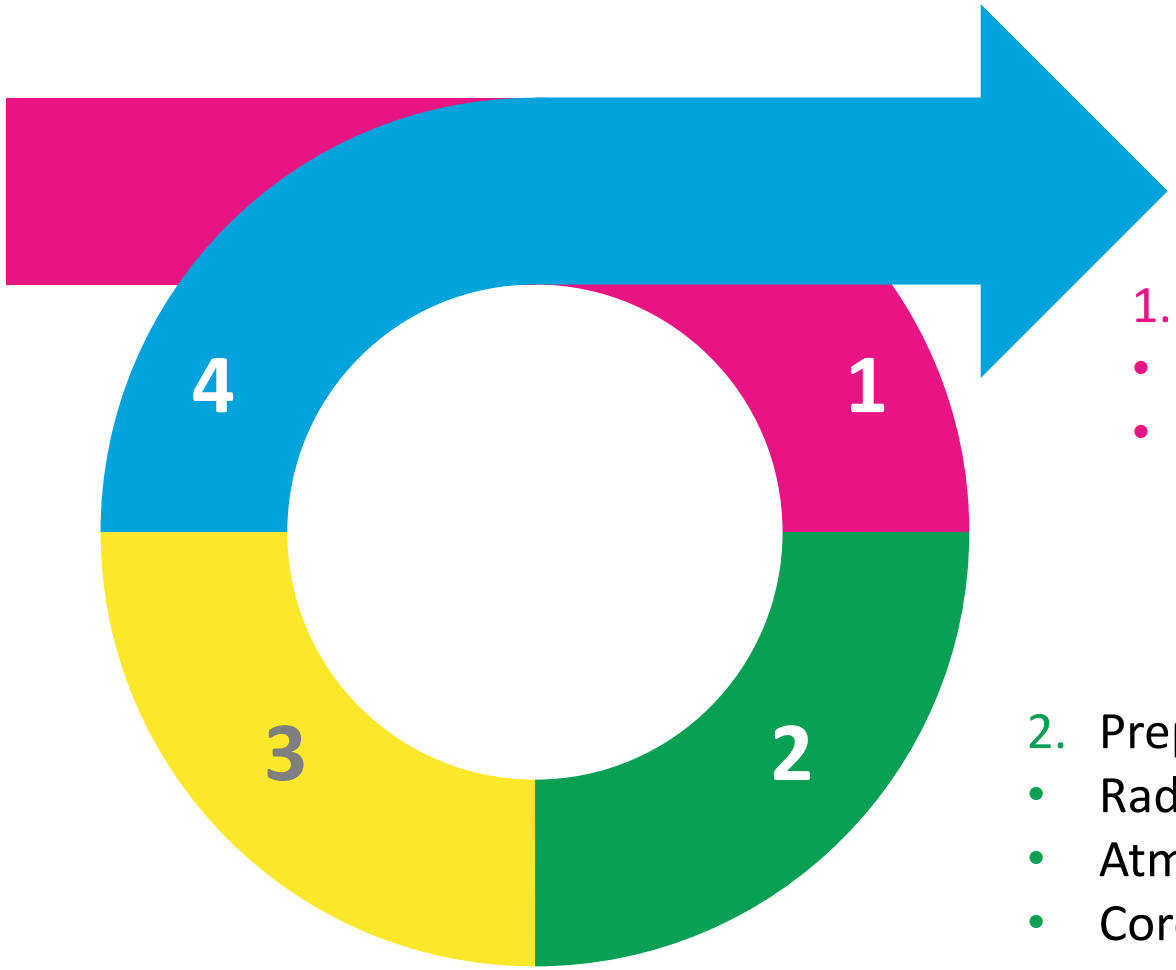
Simplified Analysis Workflow

1. Data Exploration
- Literal image interpretation
- Manual spectral analysis

2. Preprocessing / Data Prep
- Radiometric calibration
- Atmospheric correction
- Coregistration

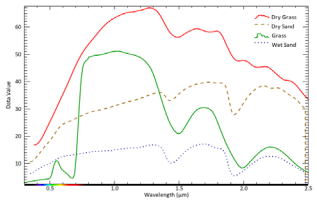
3. Data Processing
- Hardcore math

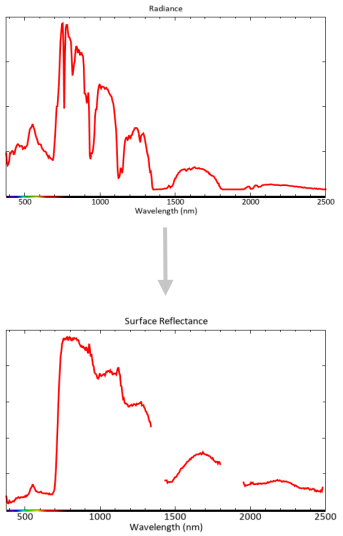
4. Results and Validation
- Image Cleanup
- Final QA/QC
- Product generation

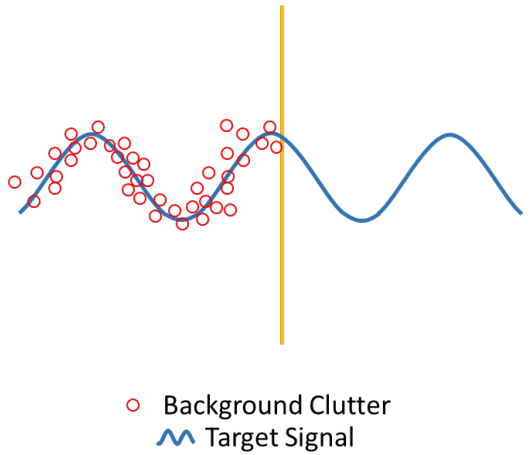


SPECTRAL SESSIONS

How ENVI Users Incorporate Open Source





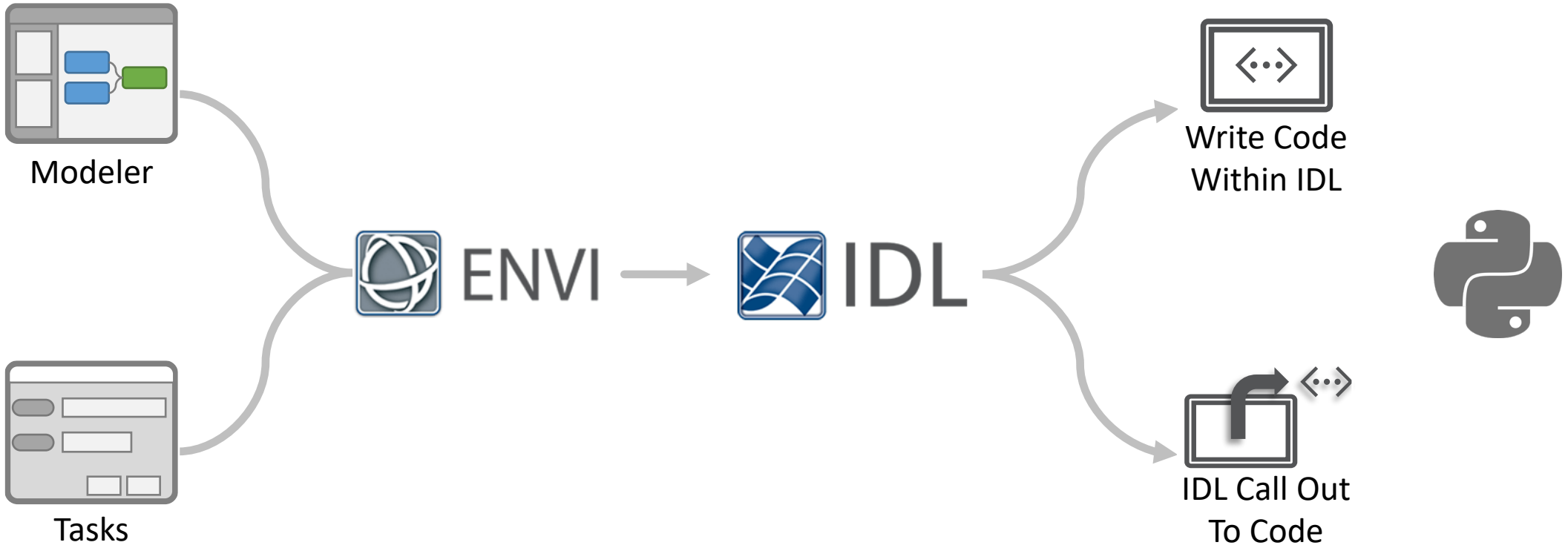


SPECTRAL SESSIONS

Entry Points

SPECTRAL SESSIONS

How Does Open Source Want to Interact with ENVI



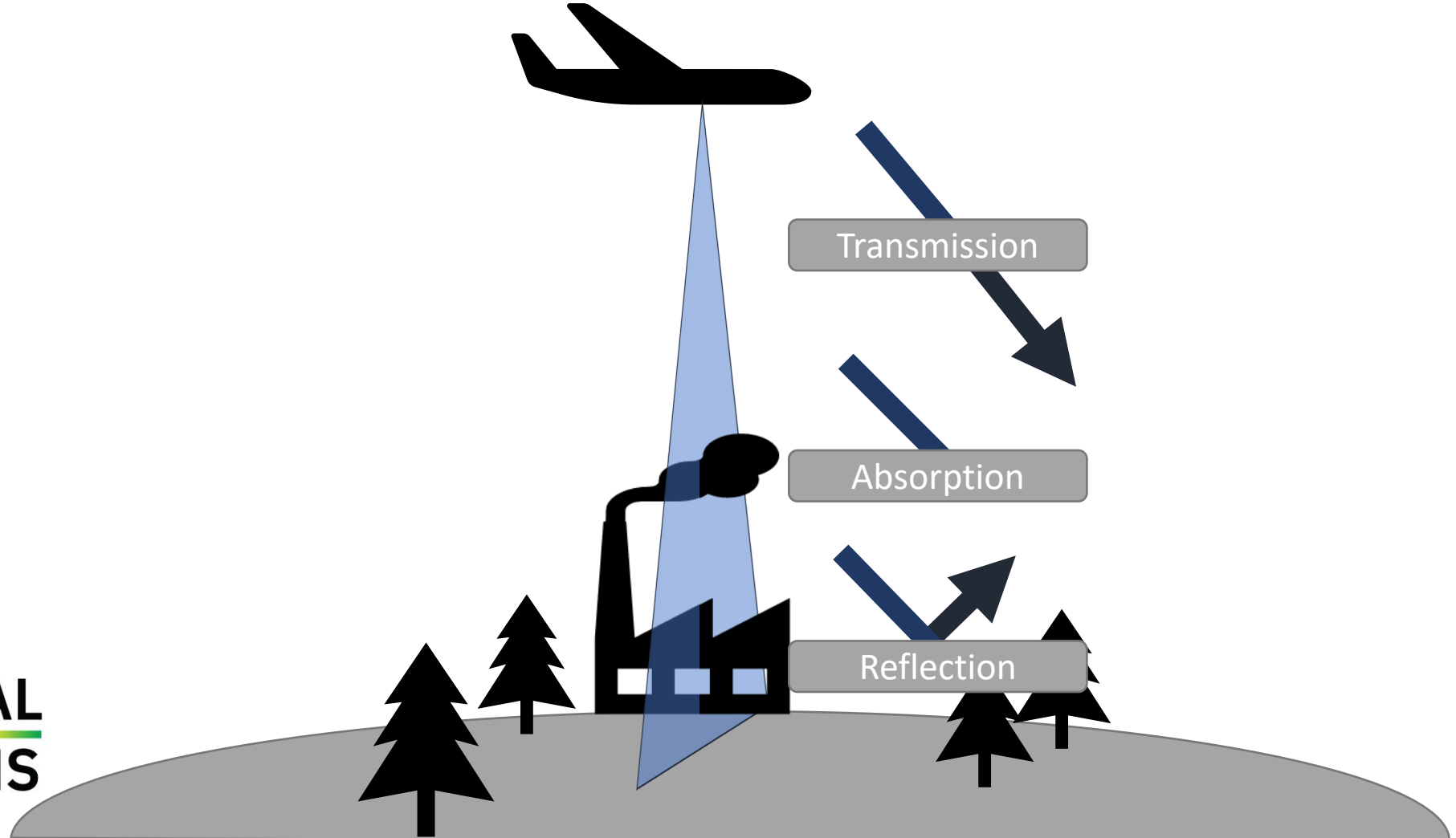
SPECTRAL
SESSIONS

Example

SPECTRAL

SESSIONS

Generalization of Methane Detection Process



SPECTRAL SESSIONS

Thank You!

THE UNIVERSITY OF UTAH®

URSA

SPECTRAL SESSIONS

Markus D. Foote (GS'20) received the B.Sc. degree in biomedical engineering from the University of Utah, Salt Lake City, UT, USA, in 2015 and the M.Sc. degree in biomedical engineering from the University of Utah in 2019. He is currently pursuing the Ph.D. degree in biomedical engineering from the University of Utah. He is a member of the Scientific Computing and Imaging Institute.

His research interests include machine learning, image processing and analysis, computational anatomy, and accelerated computing.

Philip E. Dennison received the M.A. degree and Ph.D. degree from the Department of Geography, University of California Santa Barbara, in 1999 and 2003, respectively.

He is currently professor and chair of the Department of Geography at the University of Utah. He is also the director of the Utah Remote Sensing Applications Lab. His research interests include imaging spectroscopy, remote sensing of vegetation, atmospheric trace gas retrievals, wildfire, and firefighter safety.

Fast and Accurate Retrieval of Methane Concentration From Imaging Spectrometer Data Using Sparsity Prior

Foote, M. D., Dennison, P. E., Thorpe, A. K., Thompson, D. R., Jongaramrungruang, S., Frankenberg, C., & Joshi, S. C. (2020). **Fast and accurate retrieval of methane concentration from imaging spectrometer data using sparsity prior**. *IEEE Transactions on Geoscience and Remote Sensing*, *58*(9), 6480-6492.

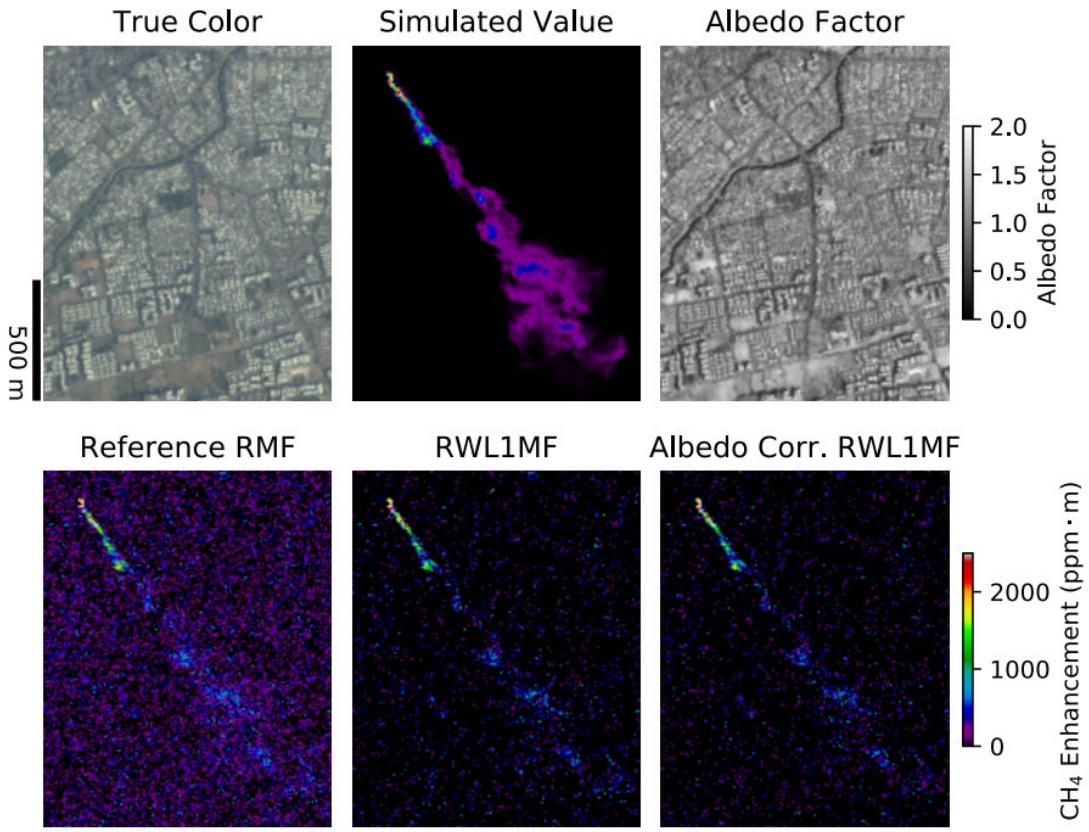
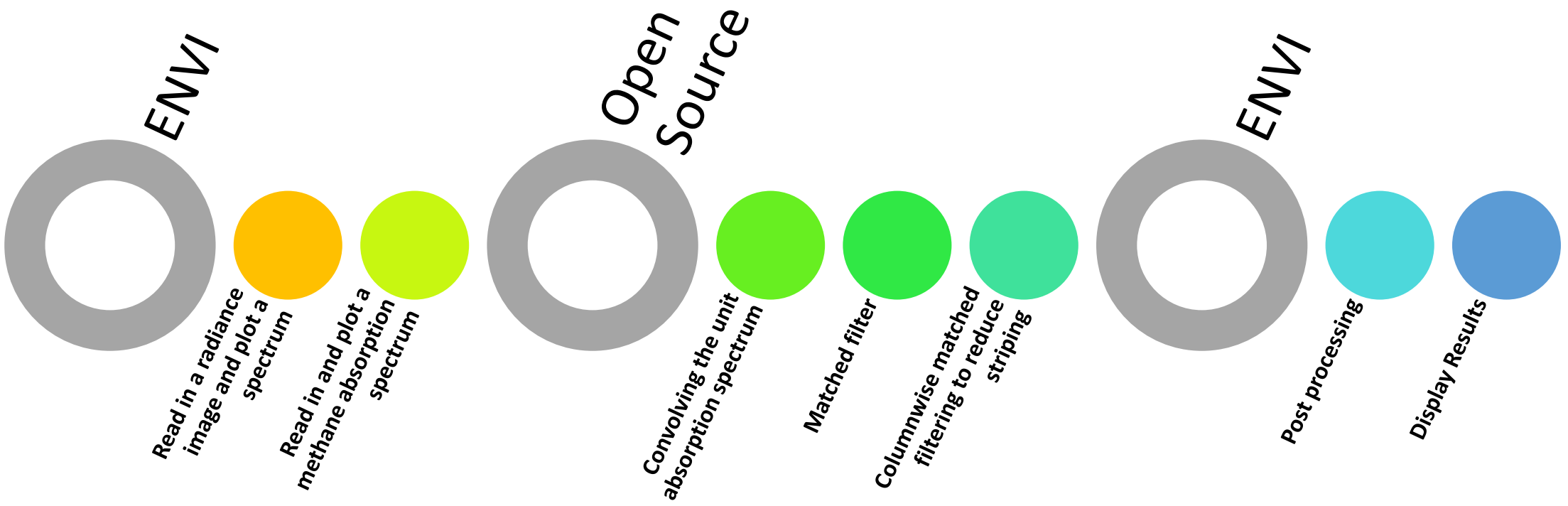


Fig. 6. Example retrievals of simulated methane plumes using each retrieval method.

SPECTRAL SESSIONS

Workflow



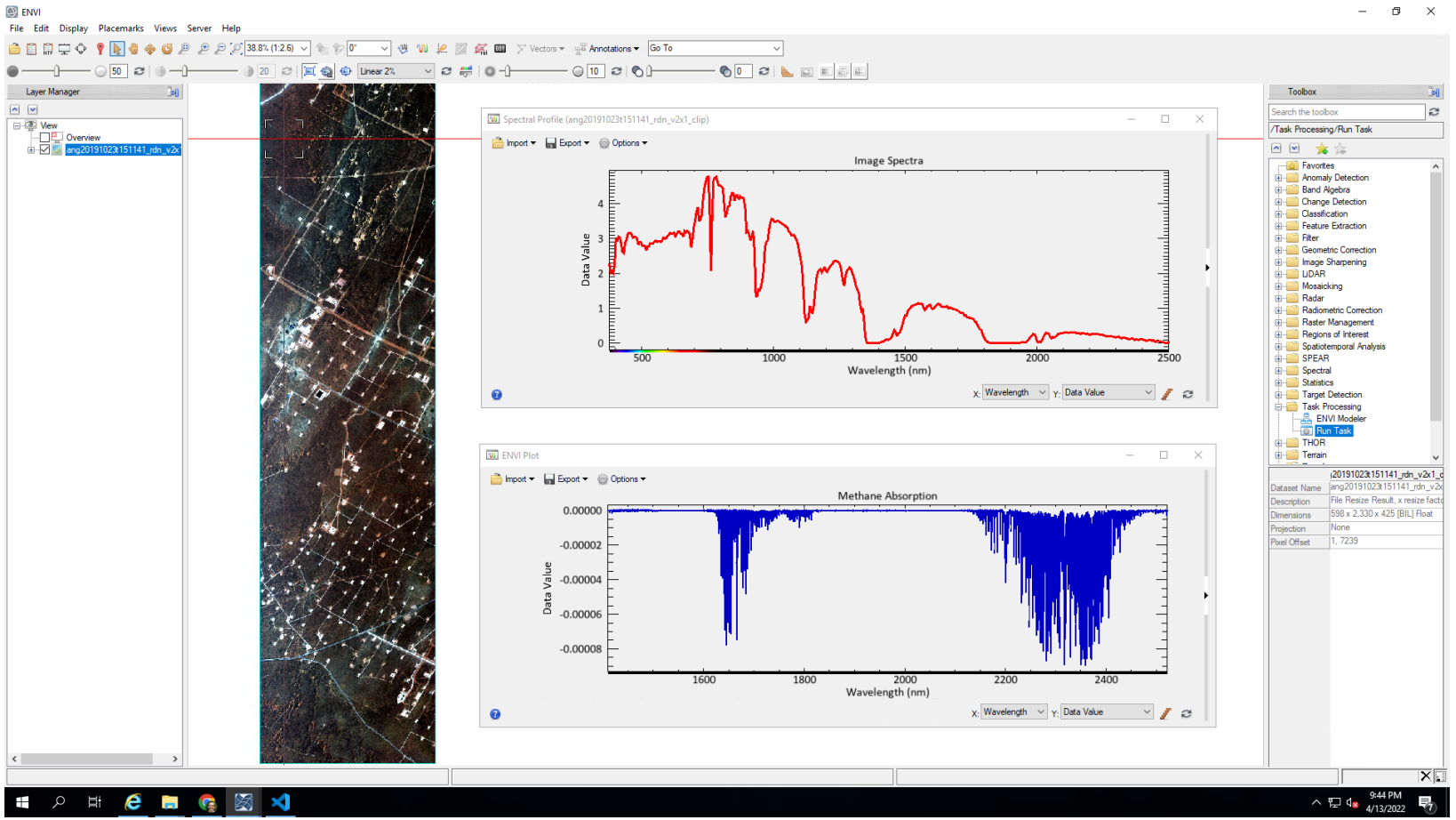
SPECTRAL SESSIONS

Block 1

ENVI

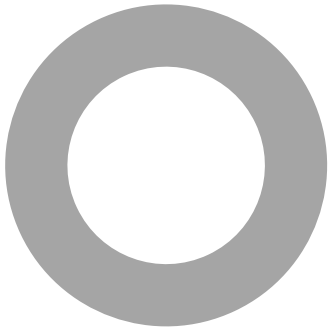
Read in a radiance image and plot a spectrum

Read in and plot a methane absorption spectrum



SPECTRAL SESSIONS

Convolving the Unit Absorption Spectrum



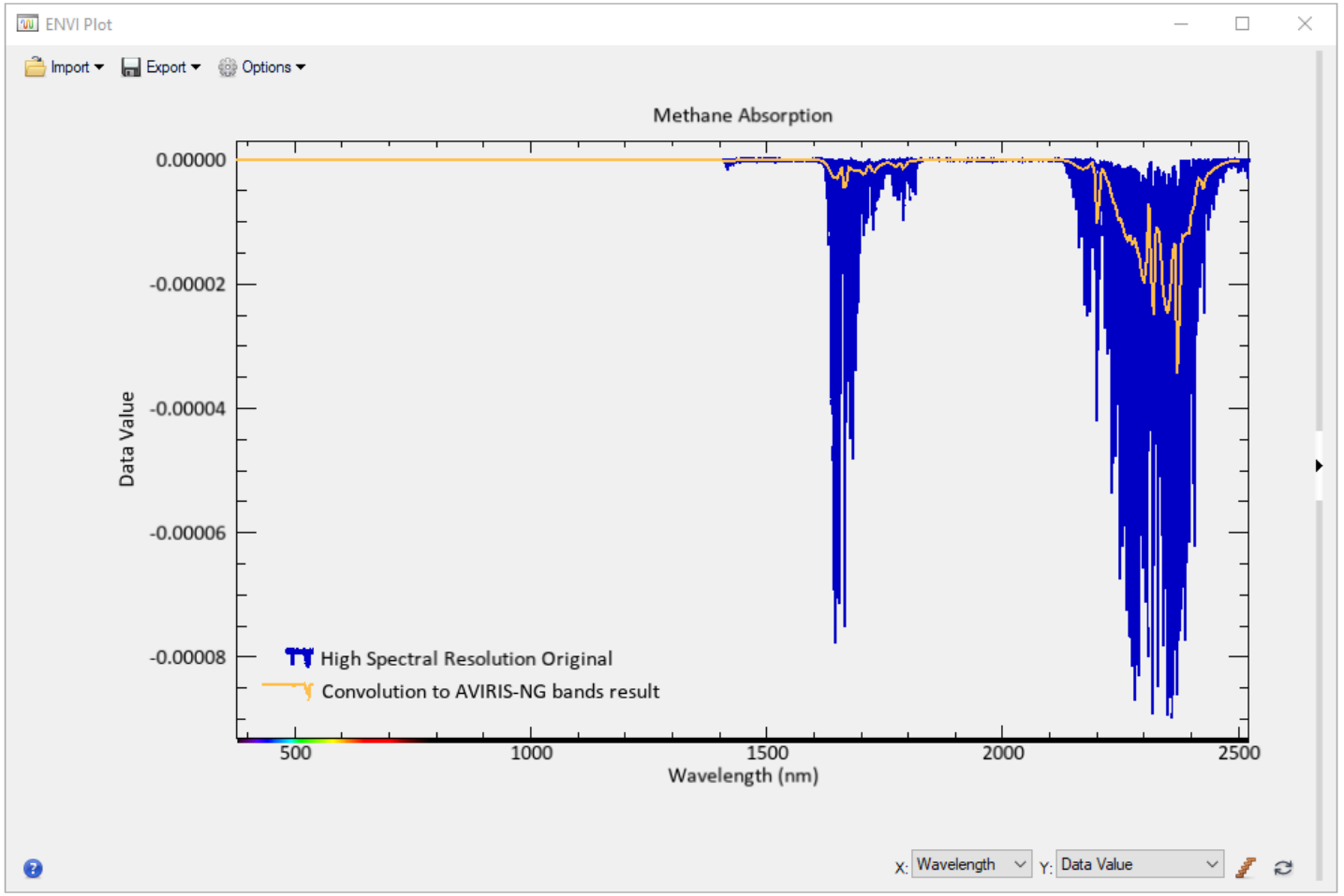
Open Source

Convolving the unit absorption spectrum

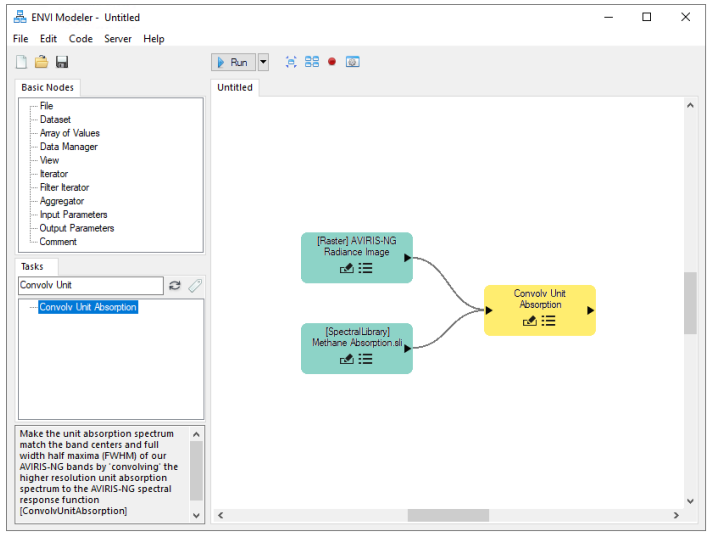
Matched filter

Columnwise matched filtering to reduce striping

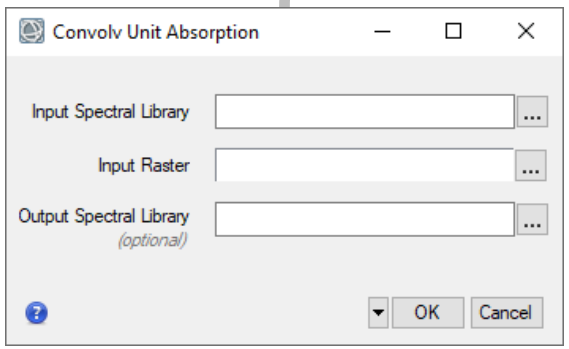
SPECTRAL SESSIONS



How Can Open Source Interact with ENVI



Modeler



Tasks

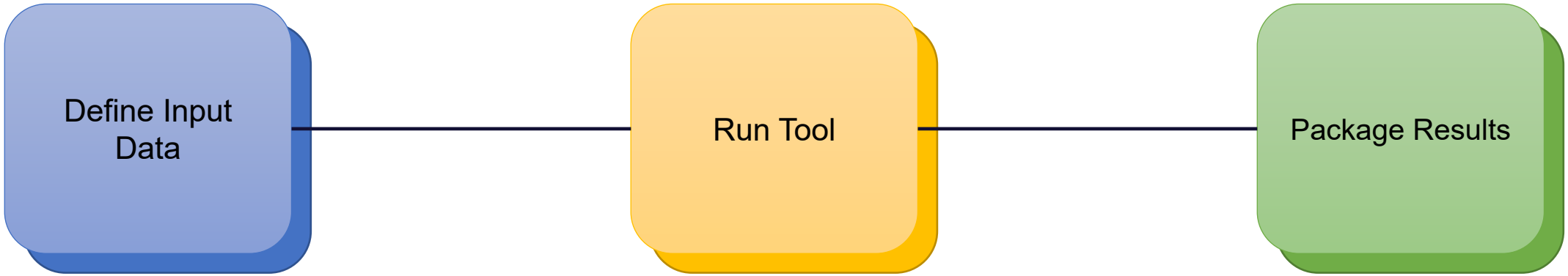
ENVI

IDL

Write Code Within IDL

SPECTRAL SESSIONS

General Format of Tool Wrapper



SPECTRAL SESSIONS

Convolving the Unit Absorption Spectrum (IDL Code)

```
; Wrap the Python Task
Python.Run, "def CUA(rast_fwhm, specLib_wl, rast_centers, specLib_vals):\n" + $

            ; Import Numpy
            "  import numpy as np \n" +$

            ; Convert FWHM into standard deviation
; Build the new library
            "  sigma = rast_fwhm / (2.0 * np.sqrt(2.0 * np.log(2.0))) \n" +$
; Get the file name
if N_ELEMENTS(output_filename) eq 0 then output_filename = e.GetTemporaryFilename('sli')
rast_fwhm = (input_raster.METADATA)['FWHM']
rast_centers = (input_raster.METADATA)['WAVELENGTH']
output_spectral_library = EnviSpectralLibrary(output_filename)
input_spectral_library.GetSpectrum((input_spectral_library.SPECTRA_NAMES)[0])
specLib_wl = absorptionSpectrum.Wavelengths
specLib_vals = absorptionSpectrum.Spectra, WL=rast_centers, WAVELENGTH_UNITS = (input_raster.METADATA)['WAVELENGTH UNITS']

            ; Assign Gaussian function weights corresponding to each AVIRIS-NG band
            "  response = np.exp(-(specLib_wl[:, None] - rast_centers[None, :])**2 / (2*(sigma**2))) / \
            np.sqrt(2*np.pi*(sigma**2)) \n" +$
            ; Force weights to sum to 1
            "  response = np.divide(response, response.sum(axis=0), where=response.sum(axis=0) > 0, \
            out=response) \n" +$

; add the spectra
            ; Implement resampling 'convolution' as matrix multiply
output_spectral_library.AddSpectra, 'Spectra #1', uas_val_avng
            "  uas_val_avng = specLib_vals.squeeze().dot(response) \n" +$

            ; Return the convolved spectra
            "  return uas_val_avng"

; Run the Python code
uas_val_avng = Python.CUA(rast_fwhm, specLib_wl, rast_centers, specLib_vals)
```

SPECTRAL SESSIONS

Convolving The Unit Absorption Spectrum (IDL Code)

Define Inputs

```
; Get the data
rast_fwhm = (input_raster.METADATA)['FWHM']
rast_centers = (input_raster.METADATA)['WAVELENGTH']
absorptionSpectrum = input_spectral_library.GetSpectrum((input_spectral_library.SPECTRA_NAMES)[0])
specLib_wl = absorptionSpectrum.Wavelengths
specLib_vals = absorptionSpectrum.Spectrum
```

Run Tool

```
; Wrap the Python Task
Python.Run, "def CUA(rast_fwhm, specLib_wl, rast_centers, specLib_vals):\n" + $

            ; Import Numpy
            "  import numpy as np \n" +$

            ; Convert FWHM into standard deviation
            "  sigma = rast_fwhm / (2.0 * np.sqrt(2.0 * np.log(2.0))) \n" +$

            ; Assign Gaussian function weights corresponding to each AVIRIS-NG band
            "  response = np.exp(-(specLib_wl[:, None] - rast_centers[None, :])**2 / (2*(sigma**2))) /
                 np.sqrt(2*np.pi*(sigma**2)) \n" +$

            ; Force weights to sum to 1
            "  response = np.divide(response, response.sum(axis=0), where=response.sum(axis=0) > 0,   out=response)    \n" +$

            ; Implement resampling 'convolution' as matrix multiply
            "  uas_val_avng = specLib_vals.squeeze().dot(response) \n" +$

            ; Return the convolved spectra
            "  return uas_val_avng"

; Run the Python code
uas_val_avng = Python.CUA(rast_fwhm, specLib_wl, rast_centers, specLib_vals)
```

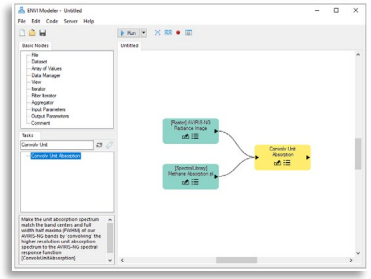
Package Results

```
; Build the new library
; Get the file name
if N_ELEMENTS(output_filename) eq 0  then output_filename = e.GetTemporaryFilename('sli')

; create the library
output_spectral_library = EnviSpectralLibrary(output_filename)

; add the wavelengths
output_spectral_library->SetProperty, WL=rast_centers, WAVELENGTH_UNITS = (input_raster.METADATA)['WAVELENGTH UNITS']

; add the spectra
output_spectral_library.AddSpectra, 'Spectra # 1', uas_val_avng
```

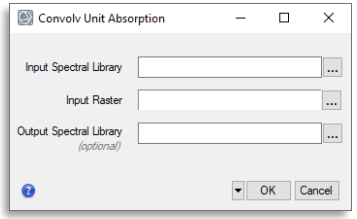
SPECTRAL SESSIONS

ENVI Custom Task Creation

- The ENVI Modeler allows ENVI Tasks to be used as building blocks for creating custom workflows.



- Tasks automatically generate a dynamic user interfaces.

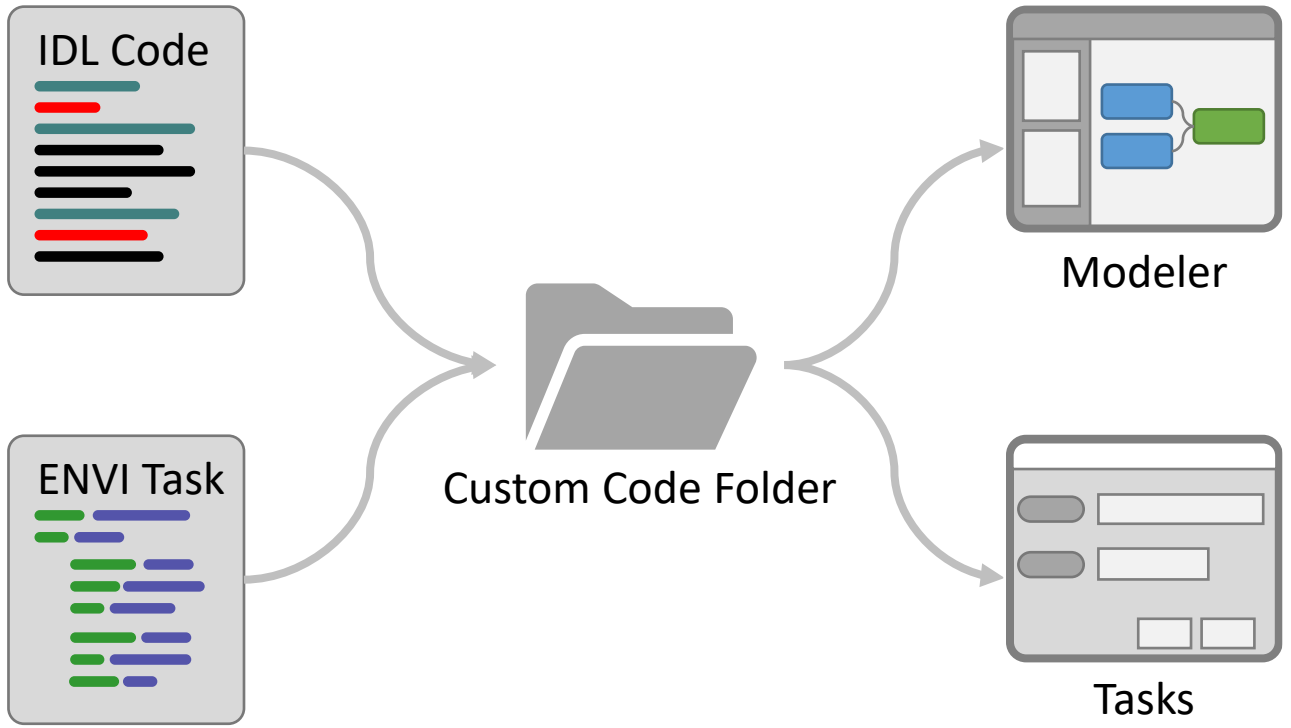


- Parameters are automatically validated.

SPECTRAL SESSIONS


```
{
"name": "ConvolvUnitAbsorption",
"base_class": "ENVITaskFromProcedure",
"routine": "ConvolvUnitAbsorption",
"display_name": "Convolv Unit Absorption",
"description": "Make the unit absorption spectrum match the band centers and full width half maxima (FWHM)
of our AVIRIS-NG bands by 'convolving' the higher resolution unit absorption spectrum to the AVIRIS-NG
spectral response function",
"schema": "envitask_3.3",
"tags": ["Methane Example"],
"revision": "1.0",
"parameters": [
      {
          "name": "INPUT_SPECTRAL_LIBRARY",
          "display_name": "Input Spectral Library",
          "type": "ENVISpectralLibrary",
          "direction": "INPUT",
          "required": true,
          "description": "Specify a spectral library from which to retrieve a particular spectrum."
      },
      {
          "name": "INPUT_RASTER",
          "display_name": "Input Raster",
          "type": "ENVIRaster",
          "direction": "INPUT",
          "required": true,
          "description": "Specify the AVIRIS-NG raster."
      },
      {
          "name": "OUTPUT_SPECTRAL_LIBRARY_URI",
          "keyword": "OUTPUT_FILENAME",
          "display_name": "Output Spectral Library URI",
          "type": "ENVIURI",
          "direction": "INPUT",
          "required": false,
          "description": "Specify a string with the fully qualified filename and path of the associated
                  OUTPUT_SPECTRAL_LIBRARY. If you do not specify this property a temporary file will be
                  created.",
          "auto_extension": ".sli",
          "is_temporary": true
      },
      {
          "name": "OUTPUT_SPECTRAL_LIBRARY",
          "keyword": "OUTPUT_SPECTRAL_LIBRARY",
          "display_name": "Output Spectral Library",
          "type": "ENVISpectralLibrary",
          "direction": "OUTPUT",
          "required": false,
          "description": "Set this property to a variable name that will contain an EnviSpectralLibrary
                  object.",
          "uri_param": "OUTPUT_SPECTRAL_LIBRARY_URI"
      }
   ]
}
```

Quick Recap of ENVI Task



SPECTRAL SESSIONS

Matched Filter

Open Source

Convolving the unit absorption spectrum

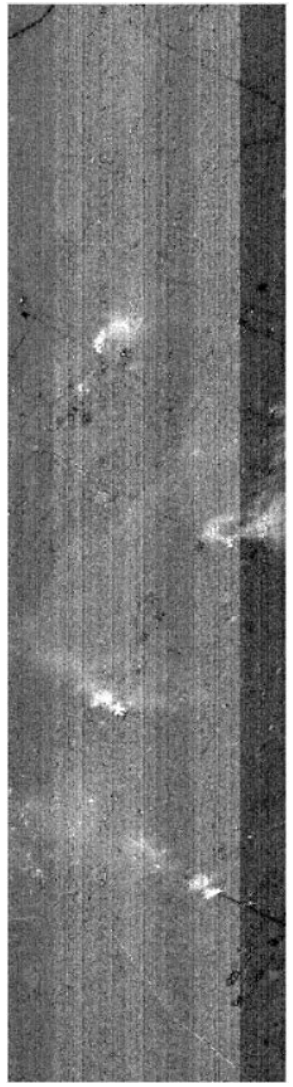
Matched filter

Columnwise matched filtering to reduce striping

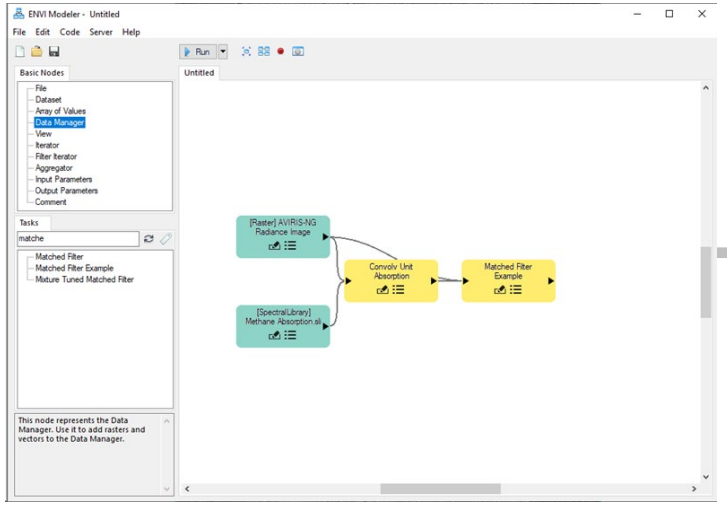
SPECTRAL SESSIONS

True Color Image

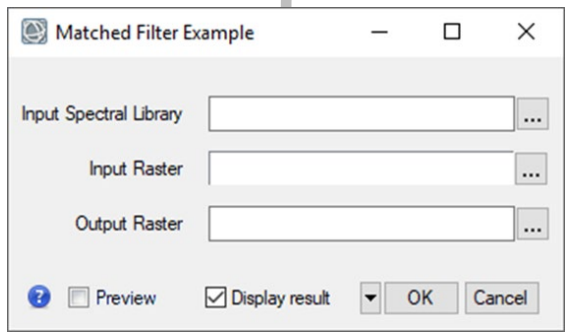
Methane Detection Image



How Can Open Source Interact with ENVI



Modeler



Tasks

IDL Call Out
To Code

SPECTRAL SESSIONS

Columnwise Matched Filtering to Reduce Striping (IDL Code)

```
; Get the data that we will be using
; Add the current directory to the python path
hdr_fname = fix_path((input_raster.AUXILIARY_URI)[0])
!null = Python.Run('import sys')
img_fname = fix_path(input_raster.URI)
!null = Python.Run("sys.path.append('"+current_dir+"')")
uas_val_avng =
output_raster = e.createraster(output_filename, result)
(input_spectral_library.GetSpectrum((input_spectral_library.SPECTRA_NAMES)[0])).Spectrum
output_raster.Save
MatchFilterExample = Python.import('MatchFilterExample')
; get the path to where this routine is located
result = MatchFilterExample.match_filter(hdr_fname, img_fname, uas_val_avng)
current_dir = ROUTINE_DIR()
current_dir = fix_path(current_dir)
```

SPECTRAL SESSIONS

Columnwise Matched Filtering to Reduce Striping (IDL Code)

Define Inputs

```
; Get the data that we will be using
hdr_fname = fix_path((input_raster.AUXILIARY_URI)[0])
img_fname = fix_path(input_raster.URI)
uas_val_avng =
(input_spectral_library.GetSpectrum((input_spectral_library.SPECTRA_NAMES)[0])).Spectrum

; get the path to where this routine is located
current_dir = ROUTINE_DIR()
current_dir = fix_path(current_dir)
```

Run Tool

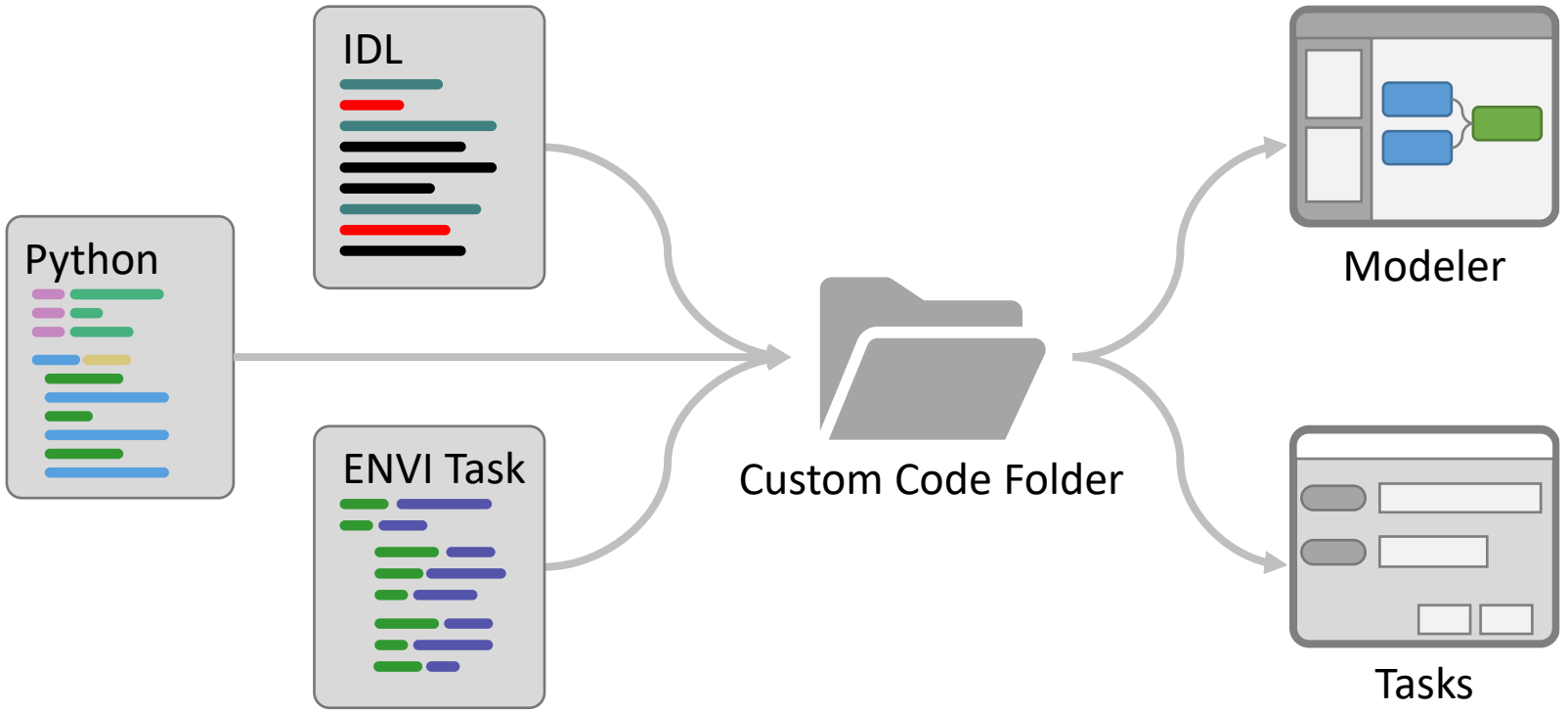
```
; Add the current directory to the python path
!null = Python.Run('import sys')
!null = Python.Run("sys.path.append('"+current_dir+"')")

; Create the IMG reference
MatchFilterExample = Python.import('MatchFilterExample')
result = MatchFilterExample.match_filter(hdr_fname, img_fname, uas_val_avng)
```

Package Results

```
; write out the new image
output_raster = e.createraster(output_filename, result)
output_raster.save
```

SPECTRAL SESSIONS

Quick Recap of ENVI Task



SPECTRAL SESSIONS

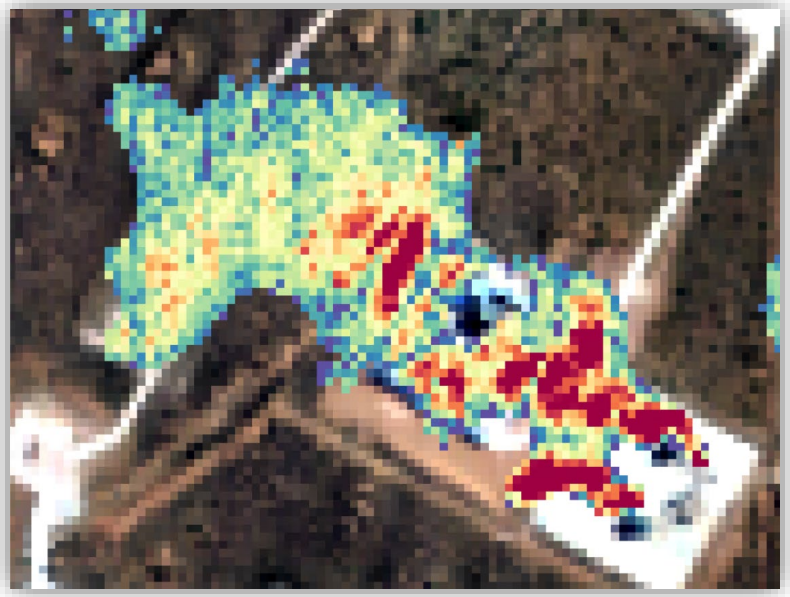
Final Display

ENVI

Post processing

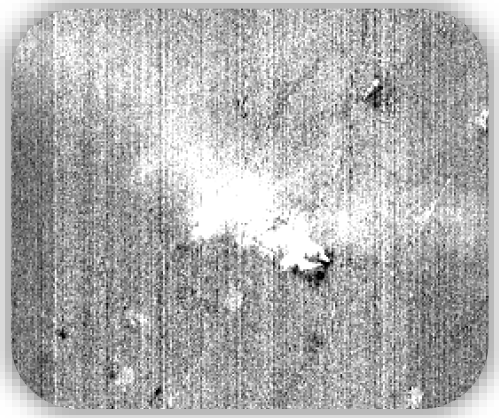
Display Results

SPECTRAL SESSIONS

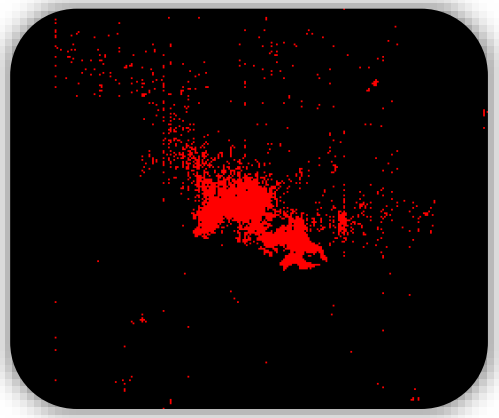


Final Display

Methane Concentration

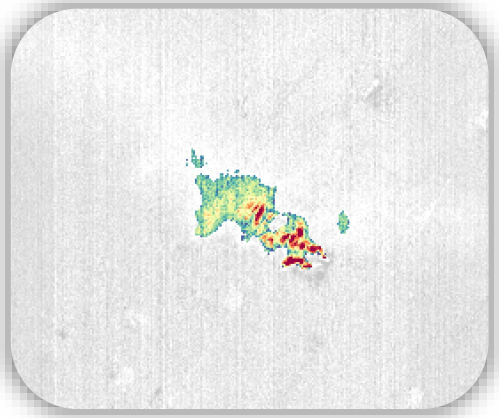


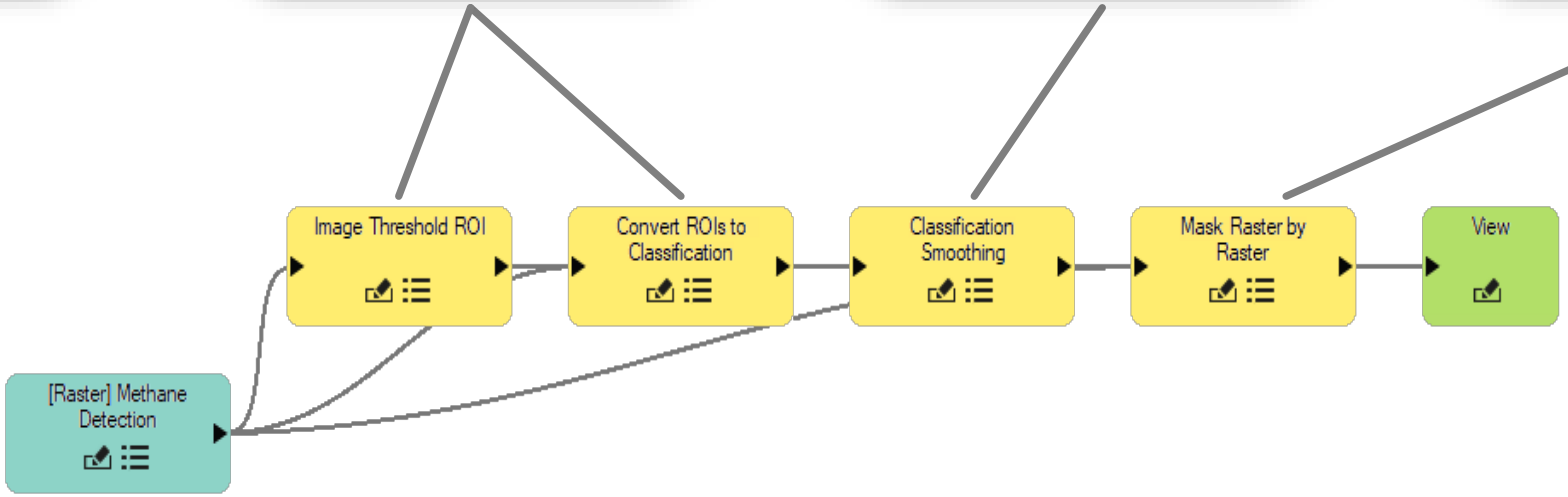
Regions of Interest (ROIs)



Filtered ROIs

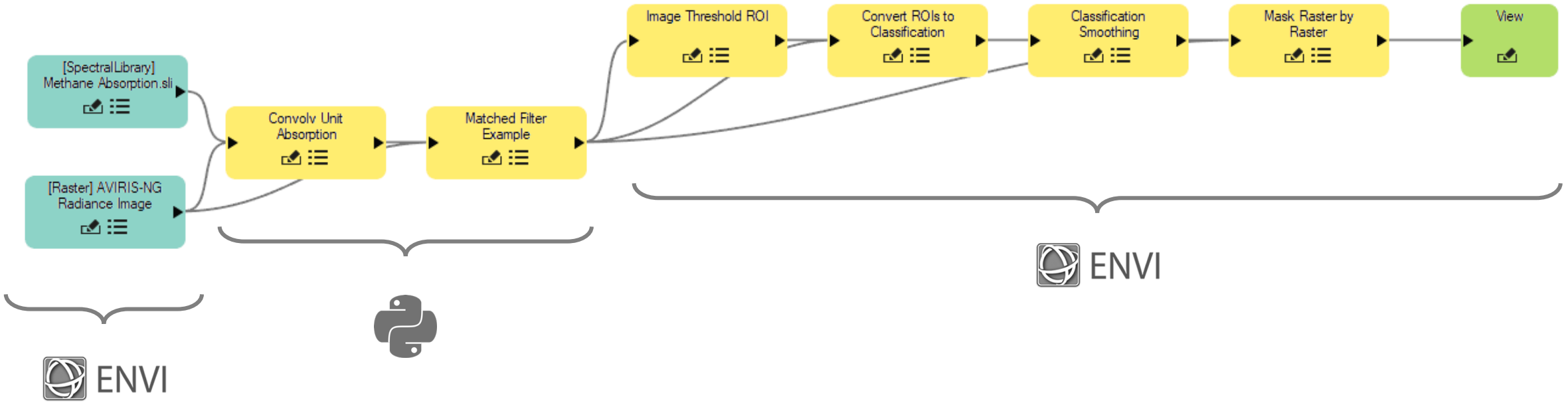
Color Sliced Results





SPECTRAL SESSIONS

Full Model



SPECTRAL SESSIONS

Questions

01

Simplified Analysis Workflow

02

ENVI Entry Points

03

Methane Example

04

General Format of Tool Wrapper

05

ENVI Task Framework

Contact Information
Austin Coates
Sales Engineering Manager
Austin.Coates@L3Harris.com