

NERC-ARF Hyperspectral Processing Practical

Task

To process a typical NERC-ARF hyperspectral flightline from Level 1 (radiometrically corrected, unmapped) to a Level 2 reflectance product then map to Level 3.

After there are some additional exercises using GDAL and calculating the Normalised Difference Vegetation Index (NDVI).

Dataset

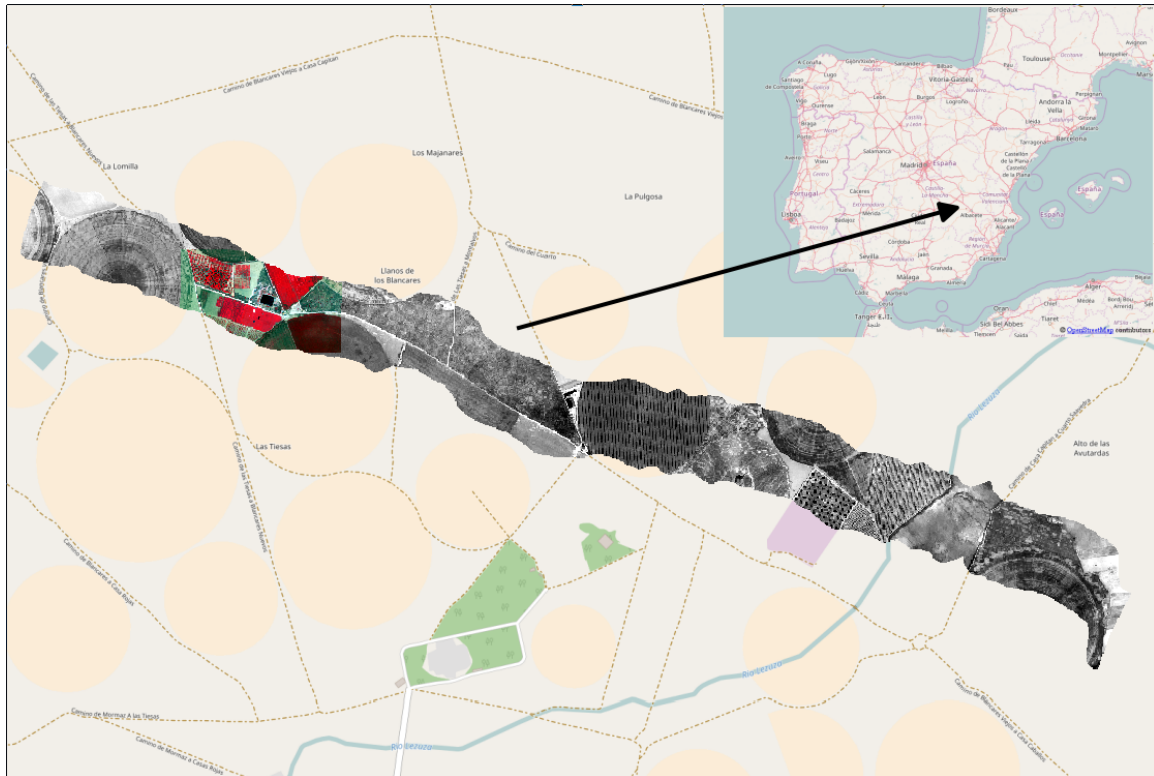
For this practical data from the Specim ASIAFenix instrument collected by the NERC Airborne Research Facility (NERC-ARF1) will be used (Figure 1). These data were flown over the Barrax site in Spain on 17th June 2015 as part of the ATSC summer school. For this practical a line 5 will be used. This line was flown at an altitude of 1400 m from south east to north west at 10:50 (GMT). The VNIR bands were spectrally binned, providing a total of 448 bands.

Only a single line has been provided for this practical, the entire dataset can be downloaded from:

http://data.ceda.ac.uk/neodc/arsf/2015/ASTC15_28/ASTC15_28-2015_168_Albacete_summer_school/

For more details on the dataset and supporting field data collection see:

<https://www.nceo.ac.uk/training/atsc-training/>



Files to be used

Level 1 data files

- f168051b.bil – level 1 file containing calibrated radiances
- f168051b.bil.hdr - header file for above
- f168051b_mask.bil - file specifying pixels deemed to be unreliable
- f168051b_mask.bil.hdr – header file for above

Navigation files

- f168051b_nav_post_processed.bil - file containing navigation information from the aircraft
- f168051b_nav_post_processed.bil.hdr - header file for above

View vector files

- fenix_fov_fullccd_vectors.bil - file to translate from the CCD array to the lens centre
- fenix_fov_fullccd_vectors.hdr - header file for above

DEM (Digital Elevation Model)

- ATSC15_28-2015_168-ASTER.dem - digital elevation model produced from ASTER data
- ATSC15_28-2015_168-ASTER.hdr - header file for above

Spectra

- 20170617-field_spec_targets.csv – Reflectance spectra of white, grey and black targets measured in the field during the flight.
- f168051b_target_spectra.csv – Average at-sensor radiance spectra of white, grey and black targets extracted from the hyperspectral line.

With the exception of the spectra, all files are ENVI Band Interleaved by Line (BIL) format files, which are binary files with a text file containing header information.

More information on these files will be given in the relevant sections.

Software

The Airborne Processing Library (APL) Suite

A set of hyperspectral processing tools developed at PML, described in the following paper:

M. A. Warren, B. H. Taylor, M. G. Grant, J. D. Shutler, Data processing of remotely sensed airborne hyperspectral data using the Airborne Processing Library (APL): Geocorrection algorithm descriptions and spatial accuracy assessment, Computers & Geosciences, Volume 64, March 2014, Pages 24-34, ISSN 0098-3004, <http://dx.doi.org/10.1016/j.cageo.2013.11.006>.

The following tools will be used for this practical:

- `aplmask` – Used to mask bad pixels from the final products
- `aplcrr` – Used to calculate the ground position of each pixel
- `apltran` – Used to convert between different projections
- `aplmap` – Used to create a georectified image using a level 1 file and positions calculated by `aplcrr`

The latest version of APL can always be downloaded from: <http://nerc-arf-dan.pml.ac.uk/trac/wiki/Downloads/software> (password required – please ask).

The source is available from: <https://github.com/arsf/apl>

NERC-ARF Scripts

A set of scripts developed by NERC-ARF-DAN and available from: https://github.com/pmlrsg/arsf_tools

- `get_info_from_header.py` - Used to get wavelengths and other information from header file.
- `apply_elc_to_bil.py` – Used to apply an empirical line calibration to a BIL file.

Simple Concurrent Online Processing System (SCOPS)

Available from: https://github.com/pmlrsg/SCOPS_processing

- `bandmath.py` – A script to perform bandmaths

GDAL

The Geospatial Data Abstraction Library (GDAL; <http://www.gdal.org/index.html>). Some utilities included with GDAL are used:

- gdal_translate: http://www.gdal.org/gdal_translate.html
- gdallocationinfo: <http://www.gdal.org/gdallocationinfo.html>

Tuiview

An open source raster viewer based on GDAL and written in Python (<http://tuiview.org/>)

Software Set up

If you are using a NERC-ARF machine or the virtual machine all the software will be installed and set up ready to use. If you are not using a NERC-ARF computer or the supplied virtual machine you will need to install these packages before starting the tutorial (see separate document on setting up computer for workshop)

Data Delivery Structure

The files to be used are provided in the same structure as you will receive in your own delivery. For clarity, files not used in this demo have been removed.

This structure is detailed below:

```
|-- dem - Digital Elevation Model
|-- doc - Documents such as the data quality report
|-- flightlines
|   |-- level1b - Level 1 data and mask files
|   |-- line_information - Metadata for each flightline
|   |-- mapped - Mapped level 3 files
|   `-- navigation - Post-processed plane navigation data
|-- logsheet - Logsheet for the flight detailing all flightlines
|-- project_information - Metadata for the project
|-- screenshots - JPG images, giving a quicklook for each mapped flightline
`-- sensor_FOV_vectors - CCD pixel view vectors
```

The only relevant directories for this task are dem, flightlines/level1b, flightlines/navigation and sensor_FOV_vectors.

Further information on directory structures will be provided in the Readme for your project. An up-to-date layout description is also located at:

<https://nerc-arf-dan.pml.nerc.ac.uk/trac/wiki/Processing/FilenameConventions#Delivereddata>

Getting Started

The tutorials use a mixture of a Graphical User Interface (GUI) and Command Line Interface (CLI). The CLI requires typing commands into a terminal window rather than using the mouse to select option. Although the CLI is a more 'old school' way of using a computer it can be very powerful and allow you to automate tasks a lot easier. The commands you need to type use a *fixed width font* it is recommended you try and type them out rather than just copying and pasting. You can use the TAB key to complete file paths, type the first couple of letters of a file or path and press tab to complete. If there are multiple files with the same start you will need to type the additional characters and press tab again, once you get the hang of this using the command line will start to get more efficient.

Linux Users

If you are using a NERC-ARF computer or the virtual machine the data to be used needs to be located at:

```
~/nerc-arf-workshop/hyperspectral_practical
```

Before starting you need to open a terminal window (Applications → Terminal) and navigate to the directory the data are stored by typing:

```
cd ~/nerc-arf-workshop/hyperspectral_practical
```

Then pressing enter.

Within the practical directory create a folder to store outputs called 'outputs'. You can do this from within a terminal window / command prompt using the following command:

```
mkdir outputs
```

Windows Users

On Windows if you created a 'start_apl.bat' file when you installed APL (see separate computer setup instructions) click on this to open a command prompt. If not open a command prompt and type :

```
set PATH="C:\APL";%PATH%
```

Substituting "C\APL" to where you downloaded and extracted the APL binaries to.

Next, navigate to the data directory by typing:

```
cd C:\nerc-arf-workshop\hyperspectral_practical\
```

Changing the path to where the data are stored.

Within the practical directory create a folder to store outputs called 'outputs'. You can do this from within a command prompt using the following command:

```
mkdir outputs
```

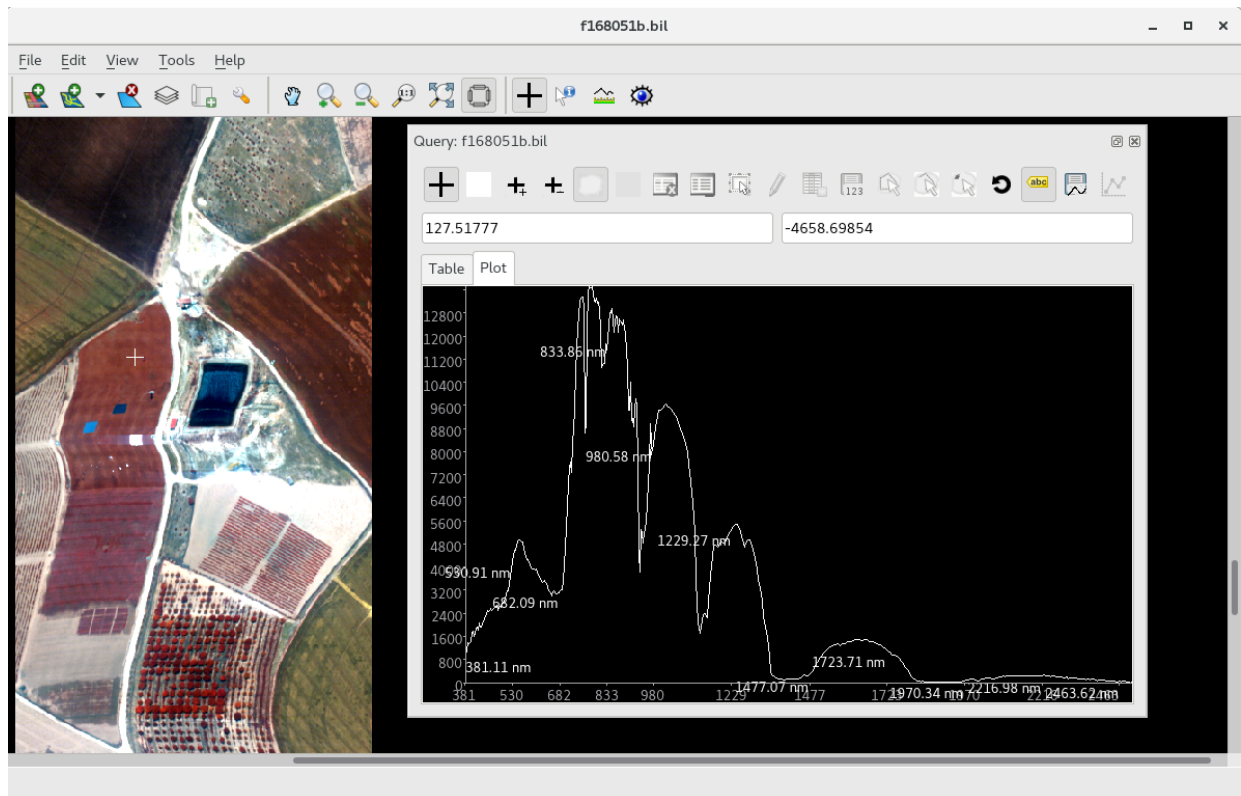
The rest of the examples in this workshop are for Linux, to use them on Windows you need to substitute the forward slashes (/) in filepaths for backslashes (\). Where commands are across multiple lines substitute the backslashes (\) for ^.

Convert at-sensor radiance to surface reflectance

Before starting processing, we will view the level1 data using TuiView. To open TuiView with the level1b data for line 1 type the following into a terminal window. If you are using Windows you will need to open the 'OSGeo4W Shell'

```
tuiview flightlines/level1b/f168051b.bil
```

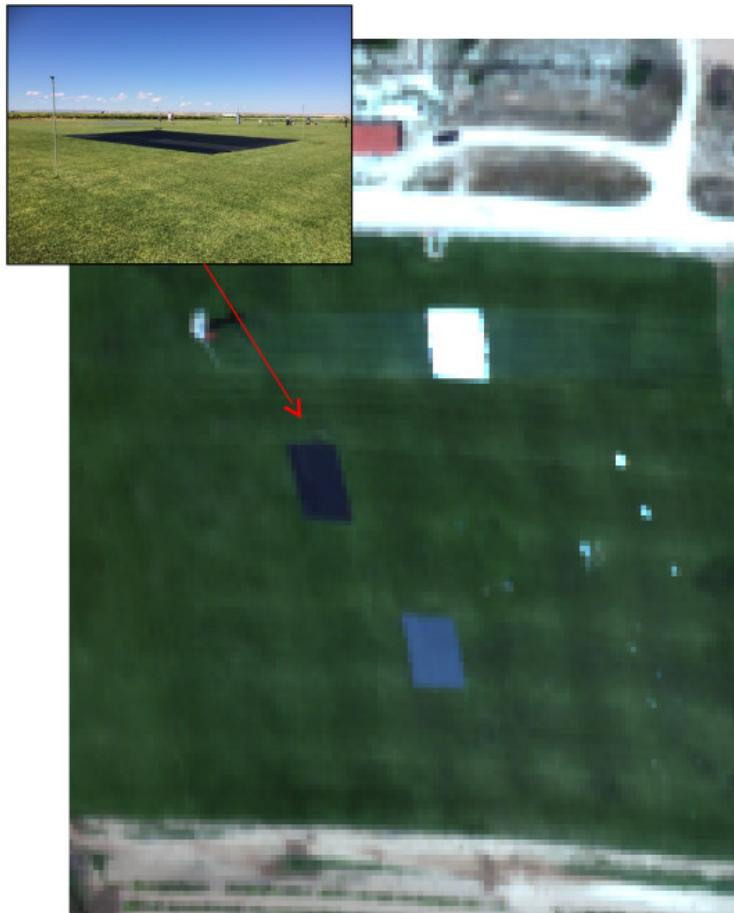
If the data are difficult to see (too dark or light) go to "Edit -> Stretch" and change the value of the standard deviation to 1.0. Click the blue down arrow in the stretch toolbar to apply to the selected layer or three blue arrows to apply to all layers. Experiment with other values to adjust the display. Select the Query button (looks like a "+") click on a pixel and select the plot tab. Click on points of the image to view the spectral profile for that pixel. Note that this only works on the top layer.



Perform an Empirical Line Calibration

As discussed in the lectures what is measured by the hyperspectral sensor on the plane and a spectrometer in the field differ. To normalise airborne data across different dates and allow comparison to field data it is often desirable to convert the hyperspectral data to units of surface reflectance. There are many ways of doing this, which will be discussed in more detail during the workshop. One method is an Empirical Line Calibration which involves creating a relationship between image radiance and surface reflectance for each band. This is often accomplished by laying out a black, grey and white target during the time of the overpass and taking measurements from them.

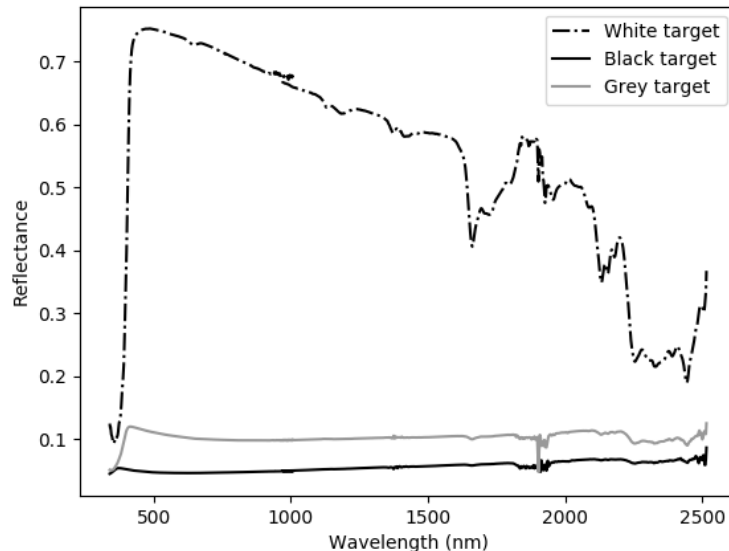
Targets were deployed for this flight and can be seen towards the bottom of the line, as shown below. With the image still open in TuiView find the targets and look at the spectra from them.



Although setting out the targets and taking measurements from them involves a lot of coordination and work, once collected it is relatively straightforward to apply the correction to the image. Various software packages contain an implementation of an Empirical Line Calibration (e.g., [ENVI](#)). For this practical a script written by NERC-ARF-DAN will be used as it doesn't require a licence.

The average spectra from each of the targets have already been extracted and saved to `spectra/f168051b_target_spectra.csv`. Multiple field spectral measurements were also taken from each of

the panels, the average has been saved to spectra/20170617-field_spec_targets.csv. A plot of the three field spectra is shown below. You may wish to open the files in Excel / LibreOffice Calc to view them.



To apply the Empirical Line Calibration, type the following command into a terminal window.

```
apply_elc_to_bil.py --image_spectra spectra/f168051b_target_spectra.csv \  
  --field_spectra spectra/20170617-field_spec_targets.csv \  
  flightlines/level1b/f168051b.bil outputs/f168051b_elc.bil
```

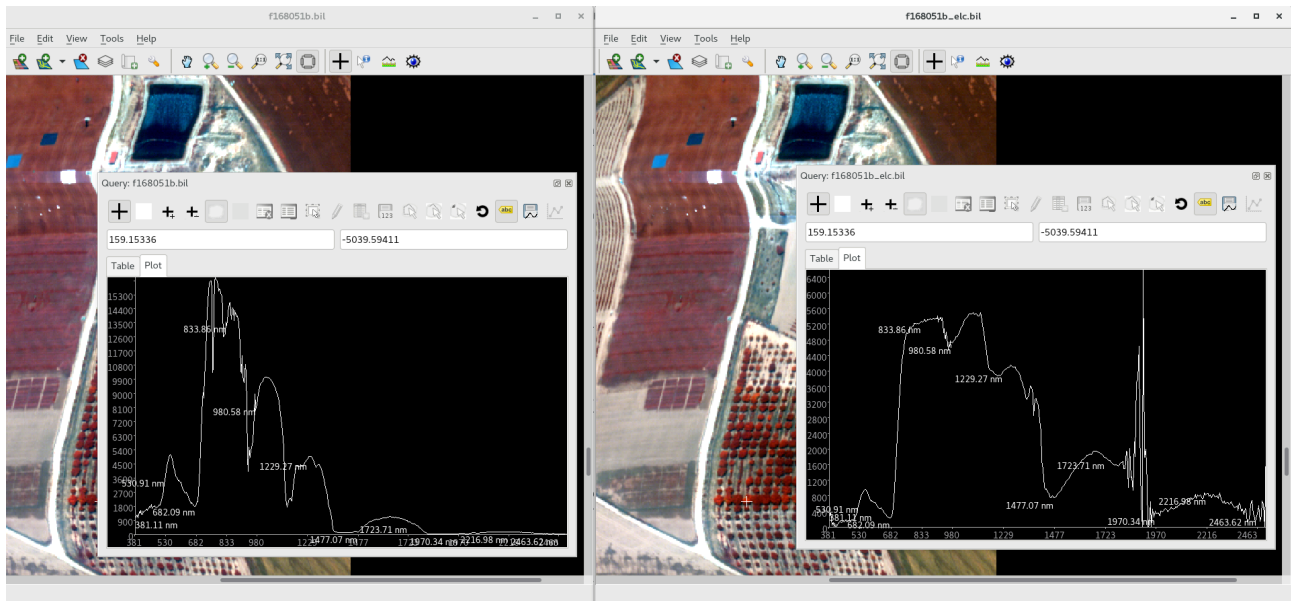
This will take a couple of minutes to run through, as it is running it will print the line it is currently on.

Once the script has finished open the output file with the original file into separate TuiView windows with the following command:

```
tuiview --separate flightlines/level1b/f168051b.bil \  
  outputs/f168051b_elc.bil
```

View the spectral profile for both of them, using the query button again. The reflectance values have been scaled by 10,000 so a value of 5,000 corresponds to a reflectance of 0.5 (50 %).

After you have looked at the spectra from different cover types close the TuiView windows.



Data Processing using APL

The following steps will take the unmapped file and map to the UTM coordinate system so it can be used with other datasets.

1. Masking bad pixels with aplmask

The first step of the APL processing is to apply the mask of bad pixels to empirical line corrected data, creating a new file with these pixels set to 0. Pixels can be 'bad' for many reasons (see mask section below).

The mask file

This is a BIL file of the same dimensions as the raw file. For every single pixel in the image (for all bands), a value is given which is the summation of the below flags:

- 0 = Good data.
- 1 = Underflows.
- 2 = Overflows.
- 4 = Bad CCD pixels.
- 8 = Pixel affected by uncorrected smear.
- 16 = Dropped scans.
- 32 = Corrupt raw data.
- 64 = Quality control failures.

A single pixel can have multiple flags. This file was created at an earlier stage in the processing chain (namely by aplcal) when various algorithms are used to identify pixels as one of the above.

Header files

Although not always explicitly mentioned in this tutorial, every time you create a new .bil file, a matching .bil.hdr file will be created. This is required as it tells programs how to interpret the BIL file. You can examine this file in any text editor and it will give you useful information such as the number of bands and the width and length of the flightline.

GUI

To open the APL GUI go to a terminal Window and type:

```
aplgui &
```

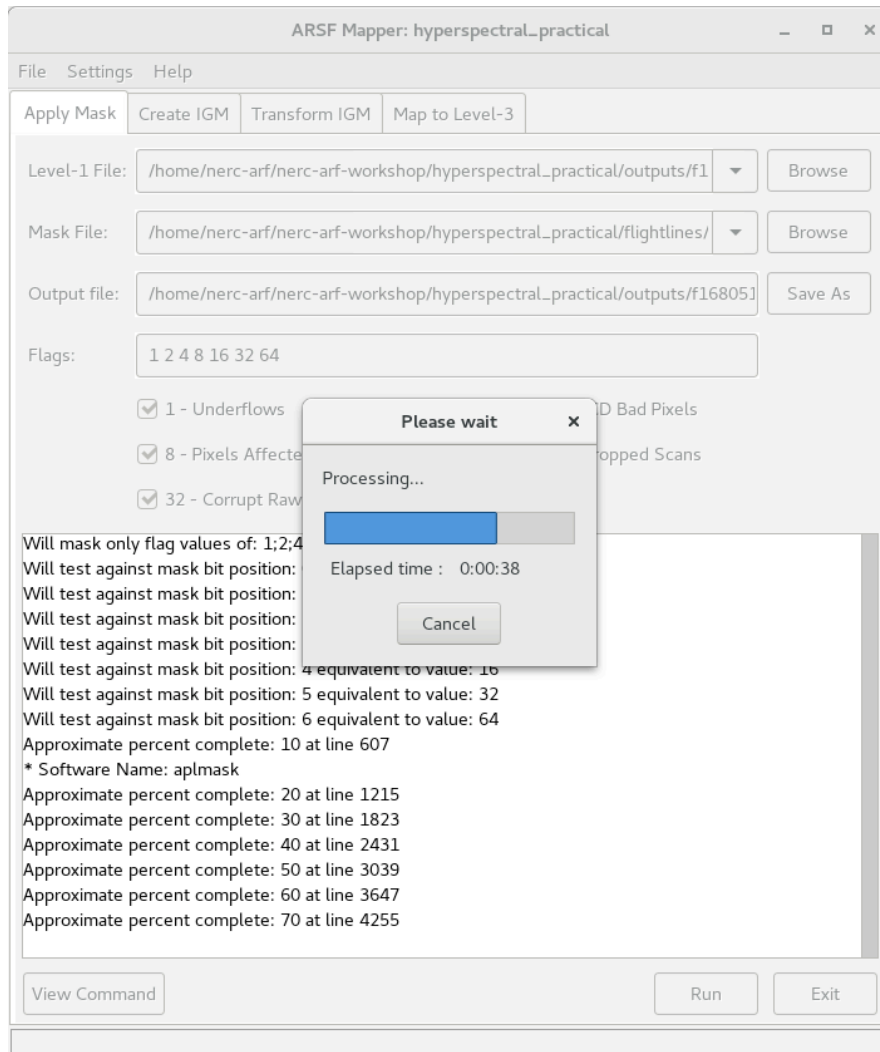
The following window will appear:



If you are not using a NERC-ARF workstation or the VM you'll need to point it at the APL executable directory so it can find the underpinning APL programs on which it is based.

a. Click File → Executable Path and point it at /usr/local/bin / C:\APL (or wherever you have saved the APL files).

1. First, we'll give it the main directory of our dataset, to make finding our input files easier in the next steps.
2. Click File → Project Directory and point it at nerc-arf-workshop/hyperspectral_practical
3. Finally, click File → Output directory and point it at your desired output location (outputs)
4. The above steps mean that future clicks of the Browse button should take you automatically to the correct directory.
5. Click Browse and find the unmapped file created from performing the empirical line calibration (outputs/f168051b_elc.bil) and the mask file (flightlines/level1b/f168051b_mask.bil)
6. Click 'Save As' for the output file and save as outputs/f168051b_elc_masked.bil
7. You can specify which mask flags to remove from the output image by ticking the appropriate boxes. Leave all of them ticked to filter out all problem pixels.
8. Click 'View Command' this shows the command which APL will run. Press 'Copy to Clipboard', we will use this to run from the command line later.
9. Click Run.
10. You should have something looking like the image below. Check for any error messages in the output.



Command Line

The command line instructions are all typed into a terminal window / command prompt. You can use the same window as the one used to start APL, you might need to press enter to get to a new line.

The first command to run is:

```
aplmask -lev1 outputs/f168051b_elc.bil -mask \
    flightlines/level1b/f168051b_mask.bil -output \
    outputs/f168051b_elc_masked.bil
```

Command break down:

- lev1: The input level 1 file
- mask: The mask file to be used
- output: The output level 1 file, with the mask applied

This is the same command run within the APL GUI. Therefore, you can just paste this by right clicking within the terminal window and selecting paste.

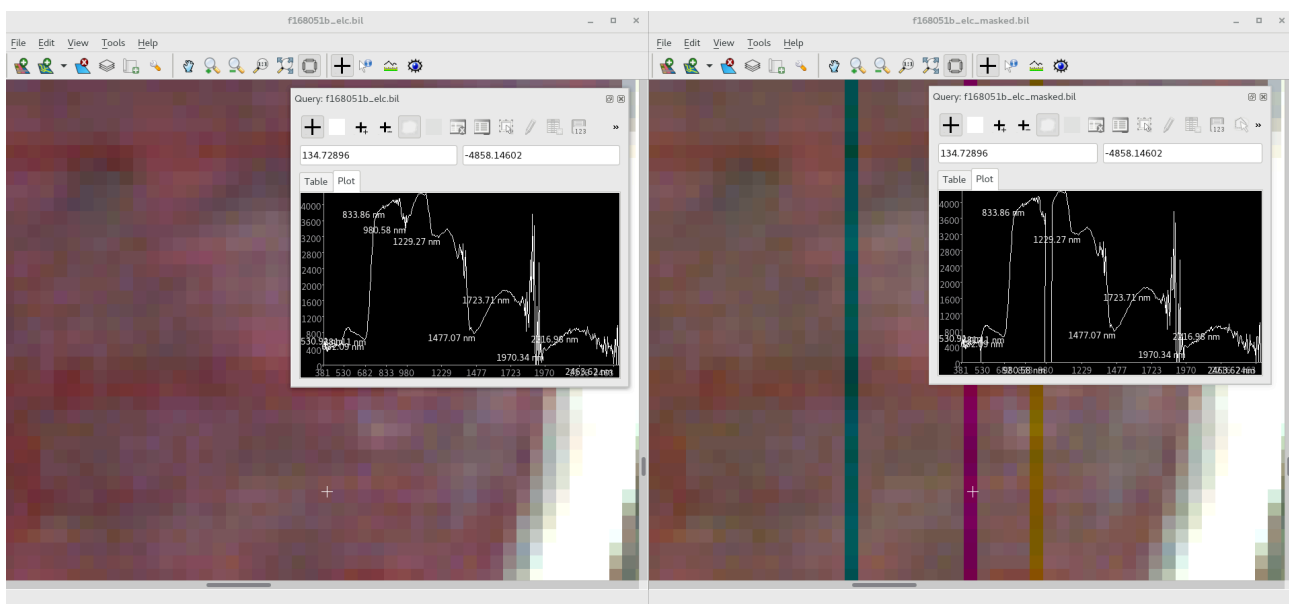
Note – the command pasted from the APL GUI will be slightly different, it will have all the flags specified, this is the default so the command can be shortened by not including it (if you have to type it out). The full path to each file is also used.

Go to a new line (by pressing Enter) to run the command.

Compare the masked files to the original by viewing in TuiView. Type:

```
tuiview --separate outputs/f168051b_elc.bil \  
outputs/f168051b_elc_masked.bil
```

Use the query button on both TuiView windows to compare spectra in the original and masked files.



2. Generate an IGM file with aplcorr

The next step uses the navigation file, the view vector file and the digital elevation file (DEM) to work out the real world ground position for each pixel in the level 1 file.

The IGM file

The Input Geometry file is a two-dimensional binary image designed to hold the geographical position of each pixel in the level 1 file.

The navigation file

This is a binary BIL file which contains position and attitude information for the plane for the duration of the flightline. The file's "bands" in this case are 7 variables: time, latitude, longitude, height, pitch, roll and

heading. There is a separate file for each instrument, due to the instruments differing position and orientation within the aircraft. This file was generated earlier using another tool in the APL suite, aplnav.

The view vector/field of view file

This file should not change within the lifetime of the sensor and is simply a way to translate from the CCD array to the principal point of the lens and hence to the correct position on the ground. Beyond remembering to include it, you do not need to worry about this file.

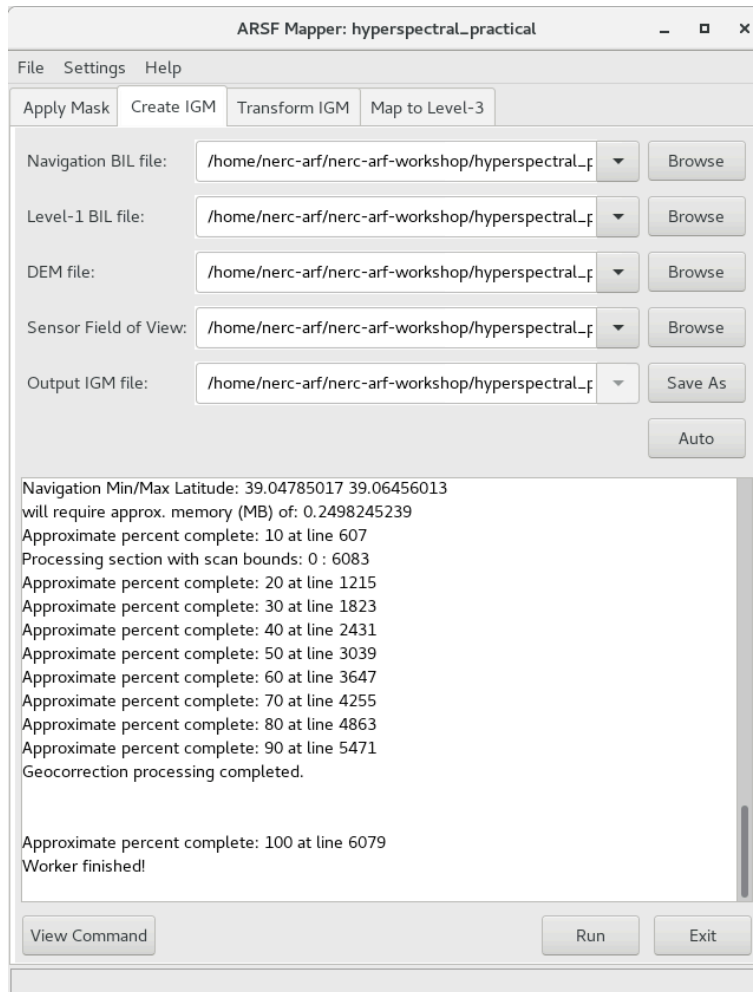
The DEM file

The Digital Elevation Model is a binary file for which each pixel has a value corresponding to its height above a given reference. Our DEM is measured against the WGS84 ellipsoid and has a pixel size of 30m. It has been produced from data from the satellite based ASTER sensor, through a collaboration between Japan and NASA (<http://www.jspacesystems.or.jp/ersdac/GDEM/E/>).

It is perfectly possible to process your data without a DEM, however where the terrain varies significantly, this will result in positioning errors.

GUI

1. Click on the "Create IGM" tab.
2. Browse to the:
 - a. navigation file: flightlines/navigation/f168051b_nav_post_processed.bil
 - b. the level 1 file: outputs/f168051b_elc_masked.bil
 - c. the DEM file: dem/ATSC15_28-2015_168-ASTER.dem
 - d. the Field of View (FOV) file: sensor_FOV_vectors/fenix_fov_fullccd_vectors.bil
 - e. the output IGM file: outputs/f168053b.igm
3. Click 'View Command' and select 'Copy to Clipboard', we will use this to run from the command line later.
4. Click Run.
5. Check output for errors.



Command Line

Remove the .igm file created using the GUI. You can do this within the terminal window by typing:

```
rm outputs/f168053b.igm
```

Type 'y' if asked for confirmation.

Generate the IGM file from the command line by pasting the command copied from the APL GUI into a terminal window or typing:

```
aplcrr -vvfile sensor_FOV_vectors/fenix_fov_fullccd_vectors.bil \
-igmfile outputs/f168053b.igm \
-lev1file outputs/f168051b_elc_masked.bil \
-dem dem/ATSC15_28-2015_168-ASTER.dem \
-navfile flightlines/navigation/f168051b_nav_post_processed.bil
```


Command break down:

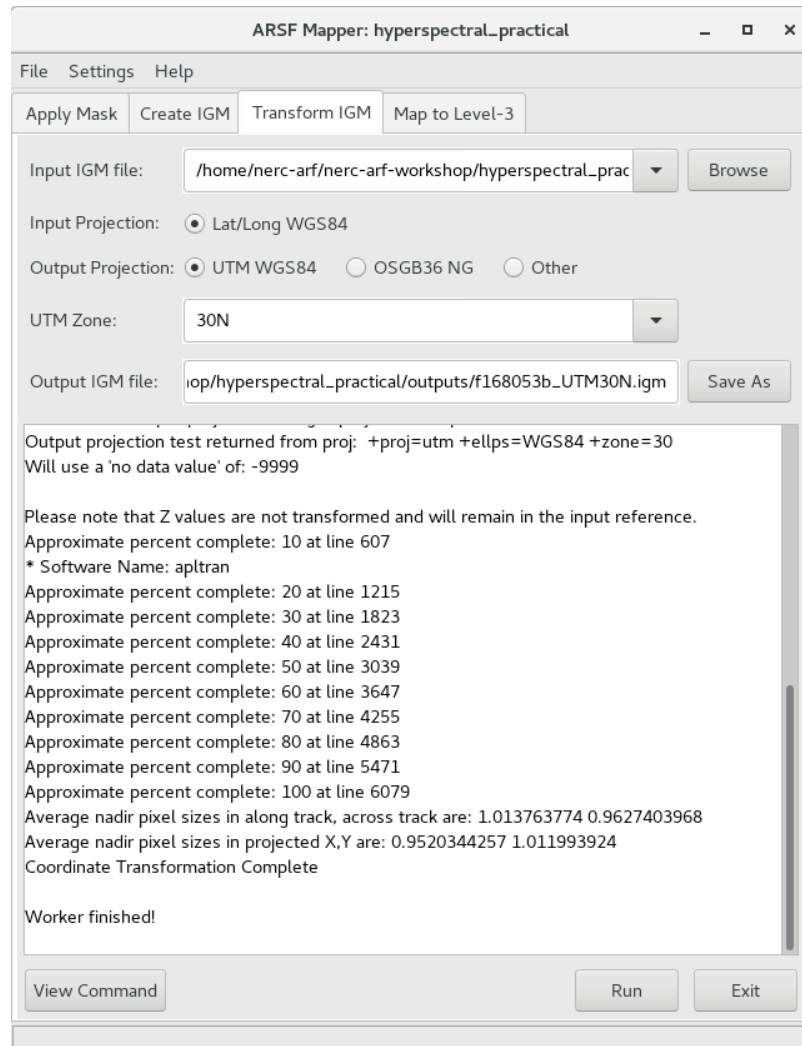
-lev1file: The input level 1 file (can be the masked or original, doesn't matter for this stage)
-igmfile: The output IGM file
-vvfile: The sensor's view vector /field of view file
-navfile: The navigation file for the flightline
-dem: The Digital Elevation Model

3. Change the projection of the IGM with apltran

The IGM output by aplcorr is always in latitude-longitude, so if you want to change the projection, for instance to match another dataset, then you should use apltran. We will be changing the projection from lat-long to UTM (Universal Transverse Mercator) to match the LiDAR data NERC-ARF also acquire.

GUI

1. Click on the "Transform IGM" tab.
2. Browse to the IGM file created above: outputs/f168053b.igm
3. Input projection should be: Lat/Long WGS84
4. Output projection should be: UTM WGS84
5. Set the UTM zone to 30N
6. Enter an output IGM file path: outputs/f168053b_UTM30N.igm
7. Click 'View Command' and select 'Copy to Clipboard', we will use this to run from the command line later.
8. Click Run. The command should only take a couple of seconds to run for the example dataset.



Command Line

Remove the .igm file created using the GUI. You can do this within the terminal window by typing:

```
rm outputs/f168053b_utm_wgs84N30.igm
```

If you copied the command from the APL GUI paste into the terminal window and press enter to run. Otherwise enter the following:

```
apltran -output outputs/f168053b_UTM30N.igm \
        -igm outputs/f168053b.igm -outproj utm_wgs84N 30
```

Command break down:

- igm: Your input IGM file created by aplcorr, in lat-long
- output: Your output IGM file (which will be in UTM zone 30 Northern Hemisphere)
- outproj: The output projection – if UTM be sure to give the correct UTM zone (30 for this flight)

4. Creating a mapped image with aplmap

The final stage is to combine the level 1 files containing the radiances with the IGM in order to create a level 3 image for which every pixel is mapped to the correct location in space, for a given co-ordinate system.

We are going to do this stage twice, creating two different mapped images.

The level 3 file

This is a binary BIL file just like the others. However, the rectangular image contained within has pixels positioned within a given co-ordinate system. The header file gives the nature of this system and the position of the image within, allowing these files to be opened in geographical software and displayed in the correct place, overlaying with other datasets if desired. As the image has to be rectangular, there will be a varying amount of empty/no data pixels in the image, resulting in a larger size compared to level 1 files.

Pixel Size

The desired output pixel size varies according to height above ground level (AGL). See this webpage for how to calculate the approximate size to use:

<https://nerc-arf-dan.pml.ac.uk/trac/wiki/Processing/PixelSize>

GUI

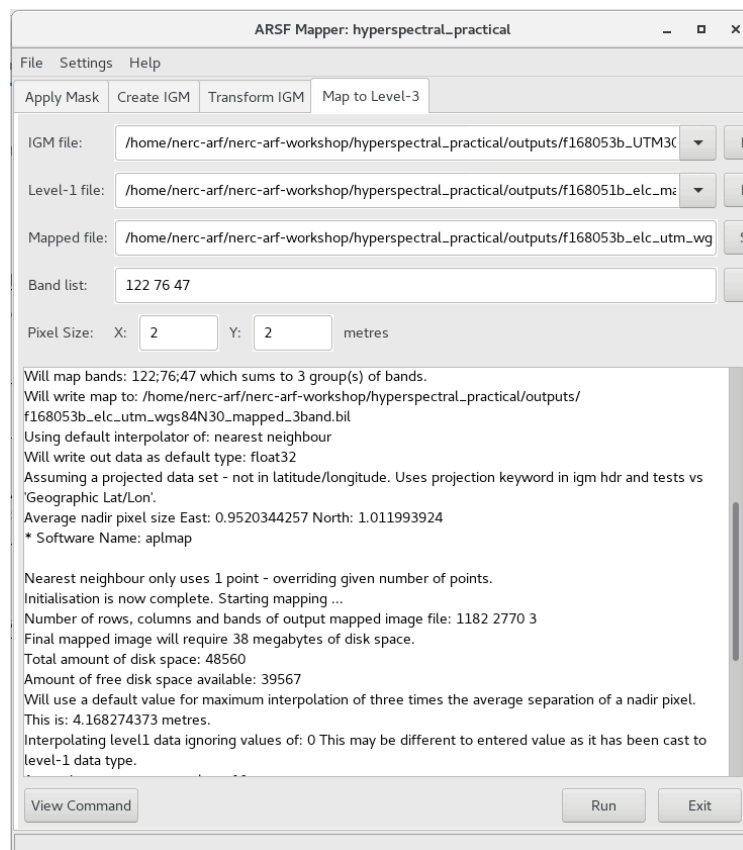
We'll run this twice, once with a subset of bands, a second time with a subset of the total area.

1. Click on the "Map to Level-3" tab.
2. Browse to the IGM file that we created with apltran: outputs/f168053b_utm_wgs84N30.igm and the masked Level 1 file: outputs/ f168051b_elc_masked.bil
3. Give a path for the output mapped file: outputs/f168053b_elc_utm_wgs84N30_mapped_3band.bil
4. We are only going to use 3 of the bands for now, so type:

124 77 47

in the band list. These correspond to NIR (~ 800 nm), Red (~ 640 nm) and Green (~ 540 nm) producing a false colour composite.

5. Enter 2 for the pixel size in the X and Y direction.
6. Click Run.



To run with a subset of the flightline, we'll need to activate the advanced options.

1. Click Settings → Show advanced controls.
2. Several new fields will appear, enter the following co-ordinates into "Area to be Mapped":

These values are min X, max X, min Y, max Y respectively.

3. Click Settings → Show advanced controls again to hide the advanced controls panel.

4. Enter ALL into the band list as we are mapping all bands this time.
5. Changed the Mapped file name to avoid overwriting:
outputs/f168053b_utm_wgs84N30_mapped_subsection.bil
6. Click Run (If you can't see the run button you might need to resize the window)

Command Line

First we'll map the whole flightline again using three selected bands.

Command:

```
aplmap -lev1 outputs/f168051b_elc_masked.bil \  
-mapname outputs/f168053b_elc_utm_wgs84N30_mapped_3band.bil \  
-bandlist 124 77 47 -igm outputs/f168053b_UTM30N.igm \  
-pixelsize 2 2 -outputdatatype uint16
```

Command Break Down:

-igm: The transformed IGM created in the above step
-lev1: The masked BIL file created by aplmask in step 1
-mapname: The output georectified BIL image
-pixelsize: Output pixel size, in metres, in the x and y direction
-bandlist: A space separated list of bands to be placed in the output image
-outputdatatype: Data type for output image, 'uint16' is a 16 bit unsigned integer, the same as the unmapped file.

Then we'll run a spatial subset, using all bands. This hasn't previously been run using the GUI

Command:

```
aplmap -lev1 outputs/f168051b_elc_masked.bil \  
-mapname outputs/f168053b_elc_utm_wgs84N30_mapped_subset.bil \  
-bandlist ALL -igm outputs/f168053b_UTM30N.igm \  
-pixelsize 2 2 -outputdatatype uint16 \  
-area 577000 578000 4323500 4324500
```

Command Break Down:

-area: A rectangular region specifying the area to process in the order min X, max X, min Y, max Y. We are using UTM coordinates so these are in metres, the X value measured from the edge of the UTM zone and the Y value from the equator.

Additional Analysis using GDAL

Unlike the subset area or band images you have created above, the level 3 mapped images supplied in a NERC-ARF delivery are of the whole flightline and for all bands. Once unzipped, these files can run into 10s of gigabytes (so for space reasons, an example file has not been included in the practical).

Change format

Command:

```
gdal_translate -of GTiff -co COMPRESS=LZW \
    outputs/f168053b_elc_utm_wgs84N30_mapped_3band.bil \
    outputs/f168053b_elc_utm_wgs84N30_mapped_3band_tif.tif
```

Command break down:

-of: Output Format - we are using GeoTiff file. For all available options see

http://www.gdal.org/formats_list.html

-co: Creation Option – specifies the creation options, these are specific to the format. For GeoTiff we use lossless compression to reduce the file size. For ENVI we can use the creation options to specify the Interleave (BSQ or BIL)

final two arguments: input and output filenames

You can get the size of the three band mapped files by typing the command:

```
du -h outputs/f168053b_elc_utm_wgs84N30_mapped_3band*
```

Note the difference in size between the uncompressed .bil file and the compressed .tif. For mapped files there are lots of pixels which don't contain any data, by using a compressed format we reduce the data size.

Extract pixel values for a given location

Previously you have viewed spectral profiles in TuiView. To extract data to plot in another packager for further analysis the `gdallocationinfo` command can be used:

```
gdallocationinfo -valonly -geoloc \
    outputs/f168053b_elc_utm_wgs84N30_mapped_subset.bil \
    577642 4324074
```

Command break down:

-valonly: Only print values to the screen
-geoloc: Interpret x and y values as coordinates rather than pixel and line
first argument: file name
x y: Easting and Northing of pixel to extract statistics from

This will print the values to the screen. To save as a text file we can redirect the output using '>'

```
gdallocationinfo -valonly -geoloc \
  outputs/f168053b_elc_utm_wgs84N30_mapped_subset.tif \
  577642 4324074 > outputs/extracted_values.txt
```

This text file can be opened in other programs for further analysis.

Using Bandmaths to Calculate NDVI

The Normalised Difference Vegetation Index (NDVI) is a commonly used index of vegetation productivity and is a ratio reflectance (ρ) in the of NIR and Red bands.

$$NDVI = \frac{(\rho_{NIR} - \rho_{red})}{(\rho_{NIR} + \rho_{red})}$$

For this example we will use 800 nm for the NIR channel and 640 nm for the Red channel.

Get wavelengths

The first step is to extract the wavelengths from the header file to a .csv file, which can be opened in Excel / LibreOffice. For this the 'get_info_from_header.py' script from NERC-ARF is used:

```
get_info_from_header.py -o outputs/f168051b_elc_masked_wavelengths.csv \
  outputs/f168051b_elc_masked.tif.hdr
```

Click to open this file in Excel / LibreOffice or type:

```
oocalc outputs/f168051b_elc_masked_wavelengths.csv &
```

Find the bands corresponding to approximately 800 nm and 640 nm.

Calculate NDVI

The next step is to use the `bandmath.py` command to calculate NDVI for each pixel within a hyperspectral file:

```
bandmath.py --equation "(band124 - band077) / (band124 + band077)" \  
            --ename NDVI --output_folder outputs \  
            outputs/f168051b_elc_masked.bil
```

Command break down:

--equation: equation to calculate, bands provided in the format band000

--ename: equation name to use for file, if not provided will use equation

--output_folder: folder to save output to

final argument: input .bil file

View using:

```
tuiview outputs/f168051b_elc_masked_NDVI.bil
```

Values closer to 1 will be pixels containing more productive vegetation. Note, there might be some gaps due to bad pixels being masked out.

Map NDVI

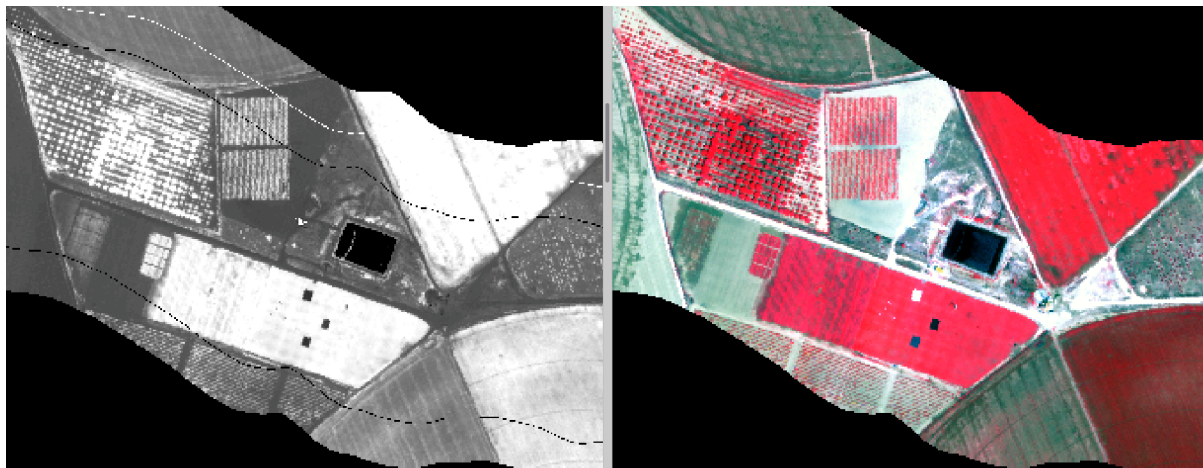
The outputs from `bandmath.py` are ENVI BIL format, as used by APL so the file created can be mapped using:

```
aplmap -lev1 outputs/f168051b_elc_masked_NDVI.bil \  
       -mapname outputs/f168053b_NDVI_utm_wgs84N30_mapped.bil \  
       -bandlist 1 -igm outputs/f168053b_UTM30N.igm \  
       -pixelsize 2 2 -outputdatatype float32
```

Or using the GUI if you prefer. Note as the values are floating point the outputdatatype is set to float32 so information isn't lost when mapping.

View the mapped NDVI and the 3 band Fenix line using:

```
tuiview --separate outputs/f168053b_elc_utm_wgs84N30_mapped_3band.bil \  
outputs/f168053b_NDVI_utm_wgs84N30_mapped.bil
```

You may wish to use this method to calculate other band metrics such as Normalised Difference Water Index (NDWI).

This practical has provided an overview of using APL to map hyperspectral data delivered by NERC-ARF and some tools you can use to create additional products which you can apply to your own data.