User-written Plugins in Spectronon

Example: Using Wien's Law to Create Heat Maps

We can use the wavelength of the peak spectral emission to estimate the temperature from the spectrum in each pixel using Wien's law:

$$\lambda_{peak} = \frac{b}{T}$$

Where b is Wien's displacement constant equal to 2898 μm·K. Solving for temperature and assuming light incident on the sensor is emitted from the fire (this analysis does not hold for reflectance data), we can estimate the temperature at each pixel from the brightest wavelength as :

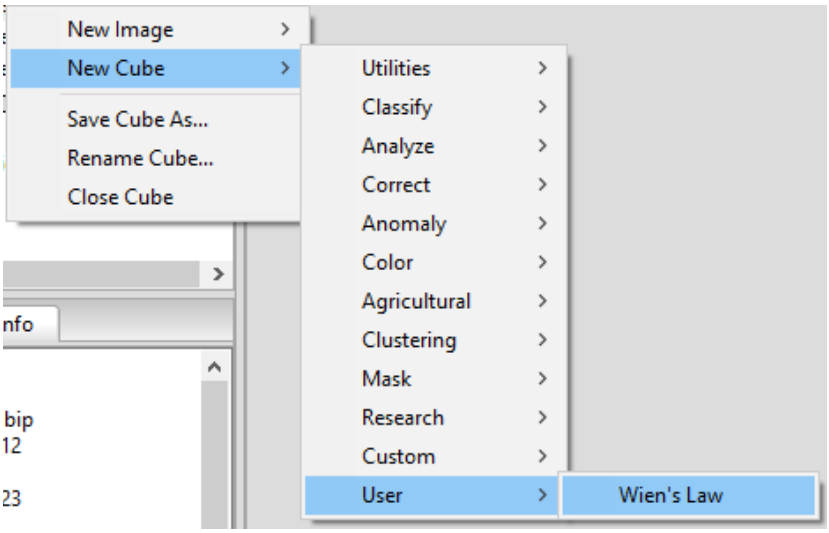
$$T = \frac{b}{\lambda_{peak}}$$

Spectronon doesn't have a bult in tool to calculate this, but Spectronon does have a Python programming interface that allows for custom analysis plugins. A simple plugin that performs these computations looks like this:

```
from src.apps.spectronon.workbench.plugin import CubePlugin
from src.lib.resonon.core.data.cube import Cube
from resononhyperspectral import Interleave
import numpy


class Wien(CubePlugin):
    label = "Wien's Law"
    tooltip = "Estimate temperature with Wien's law"

    def action(self):
        # get the raw data from the datacube, as a 3D array.  Since we request it in bip()
interleave, we know the
        # dimensions of the array are (lines, samples, bands)
        data_array = self.datacube.bip().data()

        # get the wavelengths of the datacube as a numpy array
        wavelengths = self.datacube.wavelength_array

        # For each pixel in the cube, find the index of the brightest band.  We use axis=-1
because we requested the
        # cube data in BIP interleave, so we know the first two dimensions of the data are
spatial and the last is
        # spectral.  This gives a two dimensional array where each pixel represents the index
of the band where the
        # brightest value occurs.
        brightest_band_index = numpy.argmax(data_array, axis=-1)

        # Now, convert the index of the brightest band to a wavelength by looking it up in the
wavelength array
        brightest_wavelength = wavelengths[brightest_band_index]
```

```
        # apply Wien's law to estimate the temperature in K of each pixel
        b = 2898.0 * 1000.0  # convert micrometers to nanometers
        temperature_k = b / brightest_wavelength

        # convert from Kelvin to Celsius
        temperature_c = temperature_k - 273.15

        # finally, we'll mask out parts of the cube for which the calculations are invalid.
        # Wien's law only applies to emission data - that is the portions of our image that
are actually fire.
        # Portions of the image that are dominated by a reflectance signal are not valid for
this calculation, so we'll
        # set them to zero.  For this data, we'll assume that anything with a brightest
wavelength less than 1500 mm, or
        # with a DN less than 4000 at 1600 nm is not part of the fire
        temperature_c[brightest_wavelength < 1500] = 0
        temperature_c[self.datacube.band_at_wavelength(1600) < 4000] = 0

        # Now we build and return a new datacube from the temperature data
        # datacubes are expected to be 3 dimensional, so reshape our numpy array to have
        # a 3rd dimension (a single band), then create a datacube with it:
        temperature_c = numpy.expand_dims(temperature_c, axis=-1)
        datacube = Cube.from_array(temperature_c, interleave=Interleave.BIP)

        # preserve rotation metadata from the original cube, if it has been set
        try:
            datacube.set_header_value('rotation', self.datacube.get_header_value('rotation'))
        except KeyError:
            pass

        return datacube
```

If you copy this code into a file called Wien.py and place it in the Spectronon User Plugin Directory (typically **"C:\Users\<UserName>\AppData\Local\SpectrononPro3\user_plugins\cube\user\"),** and select “Reload Plugins” from Spectronon’s file menu, Spectronon will add the custom analysis plugin to it’s menu system:



You can then run the custom analysis just like any other Spectronon Plugin! We can use the inspect tool in Spectronon to read the predicted temperatures of the fire in degrees C. The fire has a temperature range of about 1550-1650 degrees C, with the uphill part of the fire burning hotter than the downhill portion:

Here's the image with the predicted temperatues overlayed on top: