FA4: Fully Automated All-Sky Accessor and Attainer

Brett Woltz, (Moorpark College), **Farisa Morales** (Jet Propulsion Laboratory, Moorpark College)
Justin Beaurone (Jet Propulsion Lab, Moorpark College), **Vicki Truong** (Moorpark College)

Introduction

In order to study and analyze the physical characteristics of astronomical objects, like stars and their planetary systems, photometric and spectroscopic data must be retrieved from several curated databases. Unfortunately, flux (or brightness) measurements from ground- and space-based telescopes obtained at various wavelengths (from UV, visible, infrared and radio) are stored not in a homogeneous format.

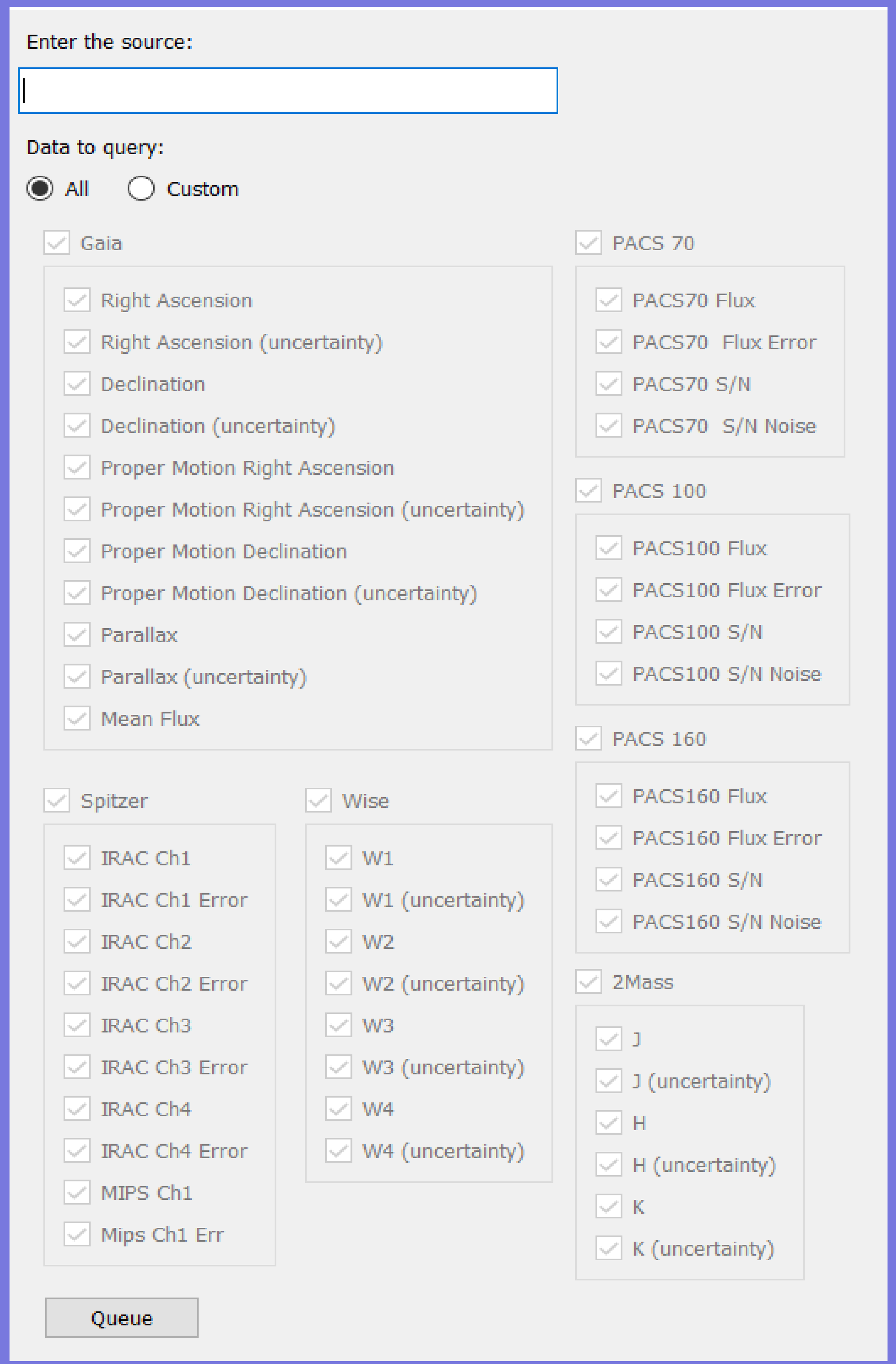
Thus, The goal of this project is to create and develop a python-based program that will query and retrieve data from these different databases, reformat it, and compile it to be used for further scientific study.

Materials and methods

The project used Python code to create a program and graphical interface to ease the retrieval of photometric and spectroscopic data from the various curated catalogs held at the IPAC (Infrared Processing and Analysis Center; *2MASS*, *WISE*, *Spitzer Space Telescope*) at Caltech, and other servers around the planet (GAIA stored at European Space Agency, and SIMBAD database stored at Strasbourg Astronomical Data Center).

The graphical user interface was created using PyQt, a Python plug-in of the cross platform GUI toolkit Qt.

The program implemented pre built packages such as "astropy" and "astroquery" to handle the data, unit conversions and query the databases.



Graphic Interface for the current version of FA4

Results

The results of this project is a computer program that can retrieve the necessary flux measurements from several telescope instruments such as those onboard *Spitzer Space Telescope* (IRAC, IRS & MIPS), *GAIA*, *Hershel Space Observatory* (PACS), and data from other sky surveys like the *Two-Micron All-Sky Survey* (*2MASS*) and the *Wide-field Infrared Survey Explorer* (*WISE*) databases.

Initially, the program worked solely using a computer terminal, and which data was queried was hard coded, meaning there was no way to pick and chose what data was retrieved. In order to change this, a simple GUI toolkit, PyQt, was adopted to make the available data selectable and to provide a graphical interface for ease use.

Some of the data from the original databases is recorded in astronomical 'magnitudes,' but needed in 'Janskys' for example. Thus, a Python function was written and implemented in order to convert the data from one format to the desired units in order to carry the scientific data analysis:

```python
# Convert flux from mag -> Jy
def zpfunc(zp, mag, err=0):
    if not err:
        return zp * 10**((-1*mag)/2.5)
    else:
        y = zp * 10**((-1*mag)/2.5)
        x = zp * 10**((-1*(mag+err))/2.5)
        return np.abs(x - y)
```

Figure. A function from the Python script to convert magnitudes to Jansky.

Finally, the queried results are saved to a text file as an ASCII table, which can be easily read by many other software. Other file formats will be supported in the future.

Conclusions

Though this student research project, I successfully created a program that can retrieve photometric and spectroscopic data from several online catalogs, and reformat it for scientific analysis.

One limitation is that the program must be stored locally on the machine in order to run it. Making the program into a web application would make it accessible on any computer that has access to the internet.

HD71722 - Notepad

File Edit Format View Help

```
\NAME='HD71722'
\RA_deg_(Gaia)=126.604762765
\RA_ERROR_mas(Gaia)=0.285843
\DEC_deg_(Gaia)=-52.807488638
\DEC_ERROR_mas(Gaia)=0.284236
\PM_RA_mass_yr_(Gaia)=-37.733081544
\PM_RA_ERROR_mass_yr_(Gaia)=0.020662033
\PM_DEC_mass_yr_(Gaia)=2.053749046
\PM_DEC_ERROR_mass_yr_(Gaia)=0.020050261
\PARALLAX_(Gaia)=14.931994
\PARALLAX_ERROR_(Gaia)=0.311323
|wavelength_microns| flux_Jy|error_Jy|         instrument|
|            double|  double|  double|               char|
|                  |        |        |                   |
|              null|    null|    null|               null|
             1.235    365.0 0.128941             2MASS_j
             1.662 4.449385  0.08121             2MASS_h
             2.159  2.94802 0.061793             2MASS_k
             3.368 1.395458 0.067707                  W1
             4.618 0.822224 0.017976                  W2
            12.082 0.140962 0.002062                  W3
            22.194 0.063385 0.002124                  W4
               nan      nan      nan       HershelPACS70
               nan      nan      nan      HershelPACS100
               nan      nan      nan      HershelPACS160
               nan 0.002519      nan               GaiaG
              3.55      nan      nan     SpitzerIRAC_Ch1
             4.493      nan      nan     SpitzerIRAC_Ch2
               nan      nan      nan     SpitzerIRAC_Ch3
               nan      nan      nan     SpitzerIRAC_Ch4
               nan  60490.0    94.44 SpitzerMIPS_Ch1_PSF
           5.21725 0.551789 0.003279      SpitzerIRS-SL2
           5.24749 0.572745 0.003317      SpitzerIRS-SL2
           5.27773 0.556096 0.003273      SpitzerIRS-SL2
           5.30797  0.54444 0.003212      SpitzerIRS-SL2
           5.33821 0.541324 0.002925      SpitzerIRS-SL2
```

Sample resultant ASCII table that is in an analysis-ready text file

Acknowledgments

Literature cited

- Astropy documentation
  <https://docs.astropy.org/en/stable/>

-PyQt5 documentation
  <https://www.riverbankcomputing.com/static/Docs/PyQt5/>