Using Python to Explore the ESA Astronomy Archives

Deborah Baines[1], Fabrizio Giordano[1], Elena Racero[1], Juan Carlos Segovia[1], Henrik Norman[1], Mattias Wångblad[1], Marcos Lopez-Caniego[1], Bruno Merín[1], Belén López Martí[1], Jesús Salgado[1], Raúl Gutiérrez[1], Pilar de Teodoro[1], Juan Gonzalez[1], Sara Nieto[1]

[1] ESAC Science Data Centre (ESDC), ESA, ESAC, Madrid, Spain

ABSTRACT

The European Space Agency (ESA) space science archives hosted at the European Space Astronomy Centre (ESAC) by the ESAC Science Data Centre (ESDC) contain many terabytes of multiwavelength data from the ESA astronomy missions: Gaia, Herschel, HST, Planck, XMM-Newton, ISO and more. In the next few years, data from three new astronomy missions will enter the archives (Euclid, JWST and PLATO), with a significant increase of the assets. While many astronomers are familiar with the archive web interfaces that are convenient for searching, visualising and downloading these data, command-line data access tools are becoming increasingly popular. The ESDC has developed python tools and astroquery modules to facilitate the scientific investigation of these data. This poster presents Jupyter notebook science cases that include accessing the ESA Gaia archive, combining multiwavelength data from various missions and visualising the data in pyESASky.

What is the Jupyter Notebook?

An interactive web-based computing environment that enables users to create notebook documents that include: - Live code - Interactive widgets - Plots - Narrative text - Equations - Images: See the example below!

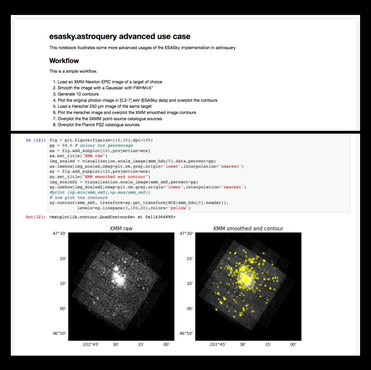
Python Libraries from the ESDC:

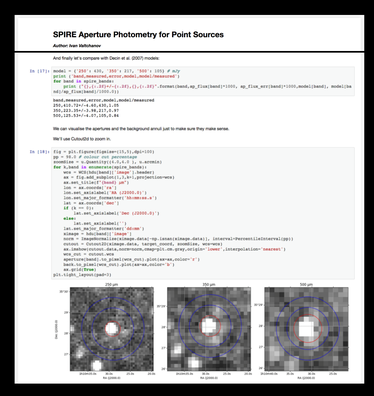
- **astro**py : **astroquery.gaia, astroquery.utils.tap, astroquery.esasky**: modules to query and access data from ESA and other archives.
- **pyESASky**: the Jupyter widget of ESASky: visualise astronomical data in ESASky within a Jupyter notebook.

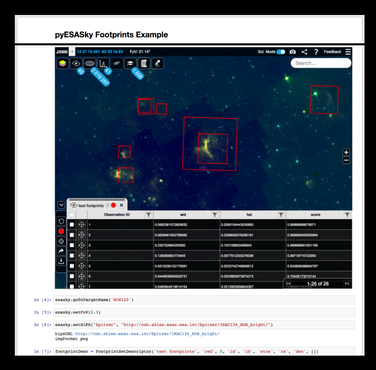
ESASky: A simple web exploration interface to facilitate data discovery and archival science lying on top of the astronomy archives:

ESASky - sky.esa.int

Notebooks available from the ESDC:







- and many more!

pyESASky instructions:

https://www.cosmos.esa.int/web/esdc/pyESASky

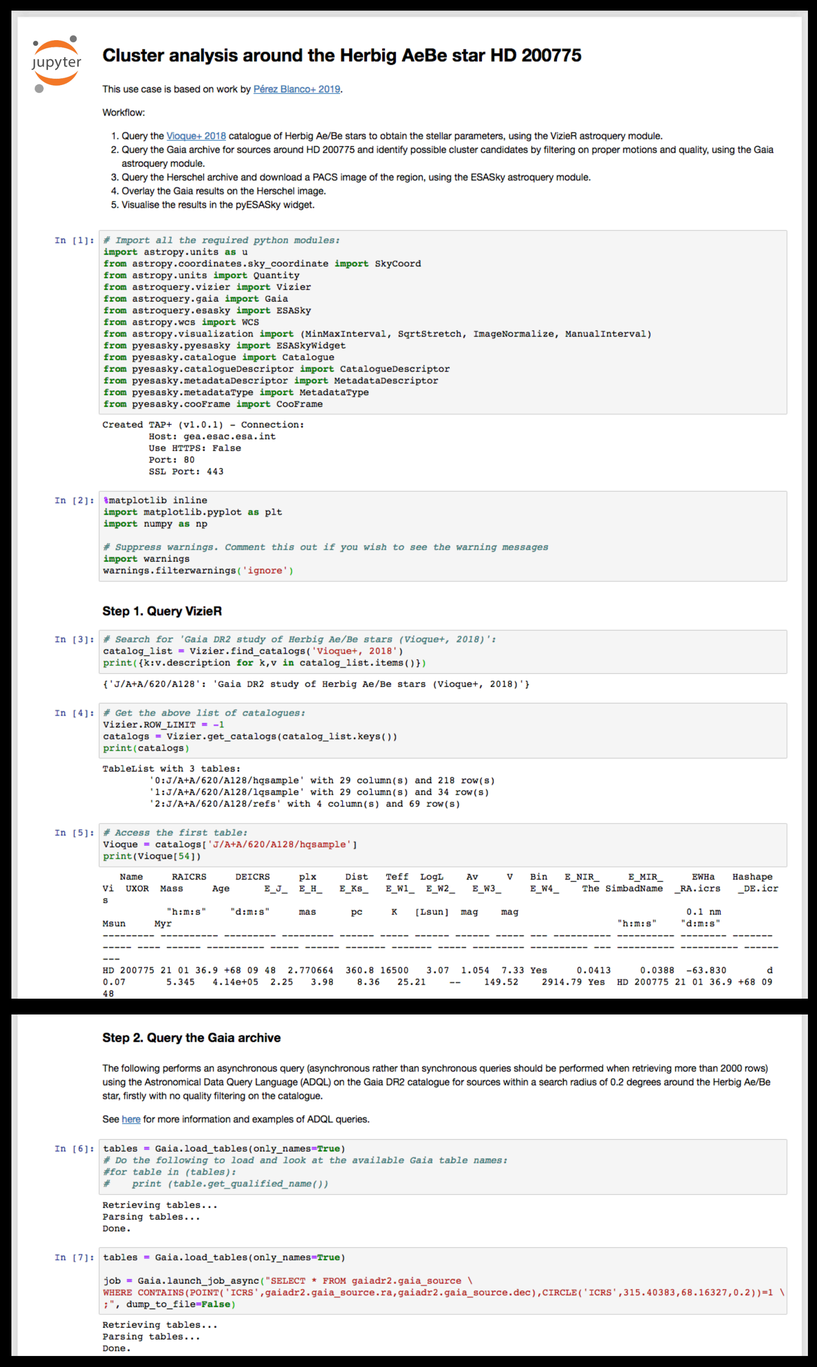
Download the notebooks here! :

https://github.com/esdc-esac-esa-int/notebooks/

Or run them in Binder: no need to install any software on your computer!

https://mybinder.org/v2/gh/esdc-esac-esa-int/notebooks/master

Jupyter Notebook Science Case Example:



Cluster analysis around the Herbig AeBe star HD 200775

This use case is based on work by Pérez Blanco+ 2019.

Workflow:

1. Query the Vioque+ 2018 catalogue of Herbig Ae/Be stars to obtain the stellar parameters, using the VizieR astroquery module.
2. Query the Gaia archive for sources around HD 200775 and identify possible cluster candidates by filtering on proper motions and quality, using the Gaia astroquery module.
3. Query the Herschel archive and download a PACS image of the region, using the ESASky astroquery module.
4. Overlay the Gaia results on the Herschel image.
5. Visualise the results in the pyESASky widget.

```
In [1]: # Import all the required python modules:
import astropy.units as u
from astropy.coordinates.sky_coordinate import SkyCoord
from astropy.units import Quantity
from astroquery.vizier import Vizier
from astroquery.gaia import Gaia
from astroquery.esasky import ESASky
from astropy.wcs import WCS
from astropy.visualization import (MinMaxInterval, SqrtStretch, ImageNormalize, ManualInterval)
from pyesasky.pyesasky import ESASkyWidget
from pyesasky.catalogue import Catalogue
from pyesasky.catalogueDescriptor import CatalogueDescriptor
from pyesasky.metadataDescriptor import MetadataDescriptor
from pyesasky.metadataType import MetadataType
from pyesasky.cooFrame import CooFrame
```

```
Created TAP+ (v1.0.1) - Connection:
        Host: gea.esac.esa.int
        Use HTTPS: False
        Port: 80
        SSL Port: 443
```

```
In [2]: %matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

# Suppress warnings. Comment this out if you wish to see the warning messages
import warnings
warnings.filterwarnings('ignore')
```

Step 1. Query VizieR

```
In [3]: # Search for 'Gaia DR2 study of Herbig Ae/Be stars (Vioque+, 2018)':
catalog_list = Vizier.find_catalogs('Vioque+, 2018')
print({k:v.description for k,v in catalog_list.items()})
```

```
{'J/A+A/620/A128': 'Gaia DR2 study of Herbig Ae/Be stars (Vioque+, 2018)'}
```

```
In [4]: # Get the above list of catalogues:
Vizier.ROW_LIMIT = -1
catalogs = Vizier.get_catalogs(catalog_list.keys())
print(catalogs)
```

```
TableList with 3 tables:
        '0:J/A+A/620/A128/hqsample' with 29 column(s) and 218 row(s)
        '1:J/A+A/620/A128/lqsample' with 29 column(s) and 34 row(s)
        '2:J/A+A/620/A128/refs' with 4 column(s) and 69 row(s)
```

```
In [5]: # Access the first table:
Vioque = catalogs['J/A+A/620/A128/hqsample']
print(Vioque[54])
```

```
    Name      RAICRS      DEICRS      plx      Dist    Teff  LogL    Av     V    Bin   E_NIR_     E_MIR_      EWHa    Hashape
Vi  UXOR  Mass      Age       E_J_  E_H_  E_Ks_  E_W1_ E_W2_  E_W3_     E_W4_     The SimbadName  _RA.icrs   _DE.icr
s                "h:m:s"     "d:m:s"     mas       pc     K   [Lsun]  mag    mag                                    0.1 nm
Msun      Myr                                                                                    "h:m:s"    "d:m:s"
--------- ---------- ---------- --------- ------ ------- ------- ------ ----- ------ ---------- ---------- --------- -------
---- ---- ------ --------- ----- ------ ------- ------- ----- ---------- ---------- --- ---------- ---------- ------
---
HD 200775 21 01 36.9 +68 09 48  2.770664  360.8 16500    3.07  1.054  7.33 Yes      0.0413     0.0388  -63.830       d
0.07       5.345   4.14e+05  2.25   3.98    8.36   25.21    --    149.52    2914.79 Yes  HD 200775 21 01 36.9 +68 09
48
```

Step 2. Query the Gaia archive

The following performs an asynchronous query (asynchronous rather than synchronous queries should be performed when retrieving more than 2000 rows) using the Astronomical Data Query Language (ADQL) on the Gaia DR2 catalogue for sources within a search radius of 0.2 degrees around the Herbig Ae/Be star, firstly with no quality filtering on the catalogue.

See here for more information and examples of ADQL queries.

```
In [6]: tables = Gaia.load_tables(only_names=True)
# Do the following to load and look at the available Gaia table names:
#for table in (tables):
#    print (table.get_qualified_name())
```

```
Retrieving tables...
Parsing tables...
Done.
```

```
In [7]: tables = Gaia.load_tables(only_names=True)

job = Gaia.launch_job_async("SELECT * FROM gaiadr2.gaia_source \
WHERE CONTAINS(POINT('ICRS',gaiadr2.gaia_source.ra,gaiadr2.gaia_source.dec),CIRCLE('ICRS',315.40383,68.16327,0.2))=1 \
;", dump_to_file=False)
```

```
Retrieving tables...
Parsing tables...
Done.
```

```
In [11]: # Run a second asynchronous query on the Gaia archive, filtering the results by quality, proper motion and parallax:
job3 = Gaia.launch_job_async("SELECT * FROM gaiadr2.gaia_source as gaia \
INNER JOIN gaiadr2.ruwe \
ON gaia.source_id = gaiadr2.ruwe.source_id \
WHERE CONTAINS(POINT('ICRS',gaia.ra,gaia.dec),CIRCLE('ICRS',315.40383,68.16327,0.2))=1 \
AND gaia.phot_g_mean_mag < 19 \
AND gaia.parallax_over_error > 10 \
AND gaia.phot_g_mean_flux_over_error > 50 \
AND gaia.visibility_periods_used > 8 \
AND gaiadr2.ruwe.ruwe < 1.4 \
AND gaia.parallax BETWEEN 2 AND 3.5 \
AND gaia.pmra BETWEEN 5 AND 11 \
AND gaia.pmdec BETWEEN -4 AND 1;", dump_to_file=False)

candidates = job3.get_results()
len(candidates)
```

```
Query finished.
```

```
Out[11]: 23
```

```
In [12]: plt.figure(figsize=(10, 8))
plt.scatter(candidates['pmra'], candidates['pmdec'], s=40, edgecolor='g', facecolor='none')
graph = plt.scatter(g['pmra'], g['pmdec'], alpha=0.9, c=g['parallax'], cmap=plt.cm.coolwarm)
cb = plt.colorbar(graph)
cb.set_label('parallax',fontsize=14)
plt.scatter(candidates['pmra'], candidates['pmdec'], s=40, edgecolor='g', facecolor='none')
plt.xlim(-20,20)
plt.ylim(-20,20)
plt.title('Gaia DR2 sources proper motion plot around HD 200775',fontsize=16)
plt.xlabel(r'pmRA',fontsize=14)
plt.ylabel(r'pmDEC',fontsize=14)
plt.legend(["HD 200775 cluster candidates"])

plt.show()
```


Step 3. Query the Herschel archive

```
In [13]: # Query ESASky for the available Herschel maps
maps = ESASky.query_object_maps('HD 200775',['HERSCHEL'])
print (maps)
```

```
TableList with 1 tables:
        '0:HERSCHEL' with 12 column(s) and 9 row(s)
```

```
In [14]: # Inspect the table
maps['HERSCHEL'].info
```

```
Out[14]: <Table masked=True length=9>
      name       dtype  format
--------------- ------- ------
   postcard_url  object
    product_url  object
 observation_id  object
observation_oid   int32
         ra_deg float64 {!r:>}
        dec_deg float64 {!r:>}
    target_name  object
     instrument  object
```

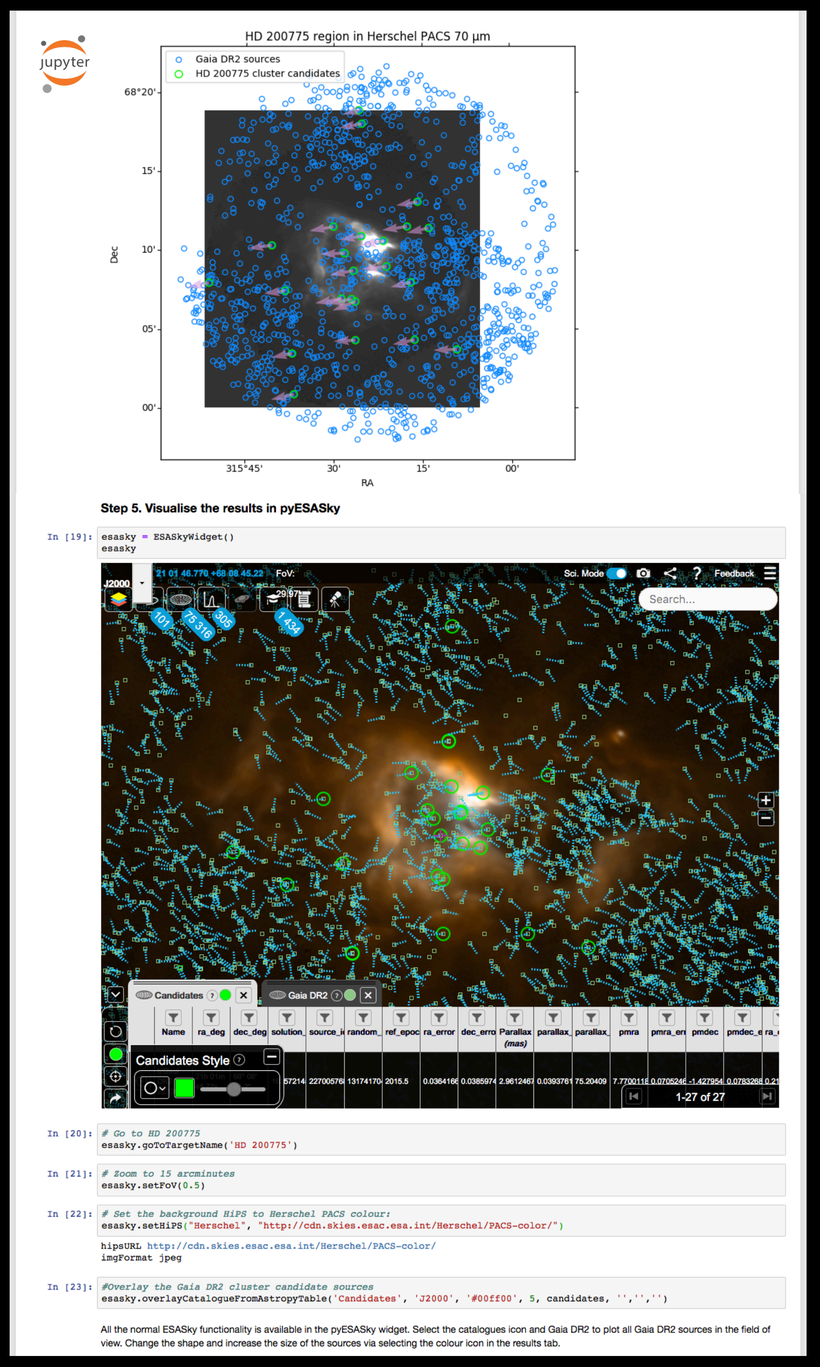
Step 4. Overlay the Gaia sources on the Herschel image

```
In [18]: # Overlaying the Gaia sources, cluster candidates and their proper motions on the Herschel image
her_image = her_hdu['image'].data

norm = ImageNormalize(her_hdu['image'], interval = ManualInterval(-0.05,0.2)) #play round with the ManualInterval numb

wcs_h = WCS(her_hdu['image'].header)
fig = plt.figure(figsize=(10,10),dpi=100)
ax = fig.add_subplot(111,projection=wcs_h)
ax.set_title("HD 200775 region in Herschel PACS 70 μm")
ax.imshow(her_image,cmap=plt.cm.copper,origin='lower',interpolation='nearest', norm=norm)
p1 = ax.scatter(g['ra'],g['dec'],transform=ax.get_transform('world'), \
                s=30, edgecolor='#1E90FF', facecolor='none', label='Gaia DR2 sources')
p2 = ax.scatter(candidates['ra'],candidates['dec'],transform=ax.get_transform('world'), \
                s=60, edgecolor='#00ff00', facecolor='none', label='HD 200775 cluster candidates')
Q = ax.quiver(candidates['ra'],candidates['dec'],-(candidates['pmra']),candidates['pmdec'],transform=ax.get_transform(
ax.set_xlabel("RA")
ax.set_ylabel("Dec")
ax.legend(["Gaia DR2 sources","HD 200775 cluster candidates"])
```

```
Out[18]: <matplotlib.legend.Legend at 0x1192f2e80>
```



Step 5. Visualise the results in pyESASky

```
In [19]: esasky = ESASkyWidget()
esasky
```



```
In [20]: # Go to HD 200775
esasky.goToTargetName('HD 200775')
```

```
In [21]: # Zoom to 15 arcminutes
esasky.setFoV(0.5)
```

```
In [22]: # Set the background HiPS to Herschel PACS colour:
esasky.setHiPS("Herschel", "http://cdn.skies.esac.esa.int/Herschel/PACS-color/")
```

```
hipsURL http://cdn.skies.esac.esa.int/Herschel/PACS-color/
imgFormat jpeg
```

```
In [23]: #Overlay the Gaia DR2 cluster candidate sources
esasky.overlayCatalogueFromAstropyTable('Candidates', 'J2000', '#00ff00', 5, candidates, '','','')
```

All the normal ESASky functionality is available in the pyESASky widget. Select the catalogues icon and Gaia DR2 to plot all Gaia DR2 sources in the field of view. Change the shape and increase the size of the sources via selecting the colour icon in the results tab.

Contact: https://support.cosmos.esa.int/esdc/ https://esdc.userecho.com dbaines@sciops.esa.int