Master 2 d'astrophysique - projet informatique Python

G.Landais

October 17, 2022

Important : refer to the "Terms and Conditions" section at the end of the document for details on the delivery of the work (in particular for the deadline for submission of monday, December 19th, 2022).

Note : Thank you to report immediately every subject misunderstanding or every technicals issues like library not installed or difficulties to install tools required for the project (refer to the section Tools required for the project).

1 Project description

Compute SED (Spectral Energy Distribution) with the photometry provided by Simbad and VizieR database.

Technical issues: integrate functions of a C-library into a Python library (Cython).
Build a user-friendly library, documented and flexible.

In addition to the code, please, provide a PDF report (2pages) explaining your implementation and a critique of the methods used (methods you have chosen or that have been imposed by the subject).

2 Detail of the project

2.1 Build a Cython interface

Integrate astrom2 library in Python using Cython (section 3.3 p.2).
The Python library must at least include the C-functions (section 3.4, p.2):

```
int search_vega_filter(char *system, char *filtername, float *lambda, float *dlambda, float *fmag)
double to_jsky(float fmag, float mag)
```

The other functions available in the C code are optional.

2.2 Build a Simbad resolver

Build a target name resolver using the Simbad database which uses the URL given in Appendix (section3.1, p.2). The resolver creates a Python object containing relevant information. Among the information we choose at least:

- the right ascension and declination
- the magnitudes (fluxes) with their errors
- the converted Jsky fluxes when the couple system/filter can be resolverd by the astrom2 library (see section 3.5 p.3)

Provide a human-readble view of the object, and a table gathering position+filter+waveband+flux(jsky)+flux_error

2.3 Extract photometry from VizieR tables

Make a VizieR conesearch which returns records of a table with a position and a radius given by user (see 3.2, p.2). Then compute fluxes in Jsky for columns that can be specified by users. Your functions must be flexible to be used for any VizieR tables. The mapping between column and the couple system/filter is configurable by the users.

Note: for the tests you can use the APOGEE or XMM tables (see 3.5, p.3). Please use small radius $< 1deg$.

2.4 Compose Simbad and VizieR photometry

For a position enrich the Simbad photometry with the photometry of a VizieR table (using a conesearch with a small radius $<= 20 arcsec$).

Note : the vizier conesearch can return a list of records having position close from the Simbad object but not exactly the same. Makes a coordinate plot (x-axis: right ascension, y-axes: declination) and a SED plot:

Instruction for SED plot construction:

- Xaxis: the lambda0 filters values
- Xerror with the dlambda filters values
- Yaxis: the fluxes values computed in Jsky (skip values that you can not convert)
- Yerror: fluxes error

Notes : You can use the positions and the APOGEE and XMM tables with the filters given in Appendices (section 3.5 p.3). Then you can compare the results with the SED VizieR viewer at:
http://vizier.cds.unistra.fr/vizier/sed/

3 Appendix

3.1 Simbad API

In this context, we exploit the HTTP Simbad API which returns a text-plain output:

`https://simbad.u-strasbg.fr/simbad/sim-id?output.format=ascii&Ident=`*TARGET*

where *TARGET* is the position name (eg.: HD1, sirius, 3C273, ...)

See also the Simbad web application: https://simbad.u-strasbg.fr/simbad/

3.2 VizieR API

We exploit the HTTP VizieR API which returns table in VOTable (XML):

`https://vizier.cds.unistra.fr/viz-bin/votable?-source=`*name*`&-c=`*center*`&-c.rm=`*radius*`&-out.all=1`

where	*name*	is the VizieR table name (e.g.: I/355/gaiadr3, II/340/xmmom2_1, III/284/allstars)
	center	the position in decimal or sexagesimal or an object name (e.g.: 0.23 +23.5 , HD1)
	radius	radius in arcmin (1deg=60arcmin, 60arcsec=1arcmin)

Catalogue example :

Name	VizieR name	URL
APOGEE	III/284/allstars	https://vizier.cds.unistra.fr/viz-bin/VizieR?-source=III/284/allstars
XMM	II/340/xmmom2_1	https://vizier.cds.unistra.fr/viz-bin/VizieR?-source=II/340/xmmom2_1

3.3 Cython

Web site : https://cython.readthedocs.io/en/latest/
Installation: pip3 install cython

In the project context you would have to pass pointer into the function - see Cython documentation.

3.4 ASTROM2 API

The M2 API is available in Moodle "astro.tar.gz" (Download: https://moodle.unistra.fr/mod/folder/view.php?id=701371).

Compilation :

```
tar -xzvf astro.tar.gz
cd astro
make
```

The compilation creates a library libastrom2.a and a program "m2filter". The functions (API) are explained in the code (c_astrom2.c).

3.5 Filters and examples

Magnitudes scale depends of the instrument. To compare the magnitudes needs to convert the values in Jsky with the function $flux = Fmag0 * 10^{(-0.4*mag)}$ (C function to_jsky(float fmag, float mag))

Simbad magnitude mapping :

Flux	System	Filter
Flux J	Johnson	J
Flux H	Johnson	H
Flux K	Johnson	K
Flux U	Johnson	U
Flux B	Johnson	B

Catalogues magnitude mapping :

Catalogue	Table name	Column name	System	Filter
APOGEE	III/284/allstars	Jmag	2MASS	J
APOGEE	III/284/allstars	Hmag	2MASS	H
APOGEE	III/284/allstars	Ksmag	2MASS	Ks
APOGEE	III/284/allstars	Mmag	Washington	M
APOGEE	III/284/allstars	T2mag	Washington	T2
APOGEE	III/284/allstars	3.6mag	Spitzer/IRAC	3.6
APOGEE	III/284/allstars	4.5mag	Spitzer/IRAC	4.5
APOGEE	III/284/allstars	5.8mag	Spitzer/IRAC	5.8
APOGEE	III/284/allstars	8.0mag	Spitzer/IRAC	8.0
APOGEE	III/284/allstars	4.5magW	WISE	W2
XMM	II/340/xmmom2_1	UVM2mag	XMM-OT	V
XMM	II/340/xmmom2_1	UVW1mag	XMM-OT	V
XMM	II/340/xmmom2_1	Umag	XMM-OT	V
XMM	II/340/xmmom2_1	Bmag	XMM-OT	B
XMM	II/340/xmmom2_1	Vmag	XMM-OT	V

Simbad Positions example :

Simbad position in APOGEE (10arcsec):
"TYC 4619-98-1" "HD 225002" "TYC 2275-512-1" "2MASS J00451128-1325229" "HD 223559 "

Simbad position in XMM (20arcsec):
"2dFGRS TGS432Z026" "2MASS J00001719+6221324" "TYC 4018-3553-1"

4 Terms and Conditions

Please, acknowledge the receipt of the project to gilles.landais@unistra.fr.

The requested work is a personal project, the code must not be shared (even partially)
The deadline for submission is monday, December 19th, 2022.

Submission modalities :

- gather all documents including code and results (png, CSV file) into a .zip file. A report-document is also required (1 or 2 pages are enough) to describe the program, the functions and (eventually) problems encountered (bugs, restrictions,...).
- Then send the archive to gilles.landais@unistra.fr. I will confirm the good reception after verification (readable files after extraction) of the contents of the archive.