VO School @ OCA- Online tools
Online tools for planetary sciences

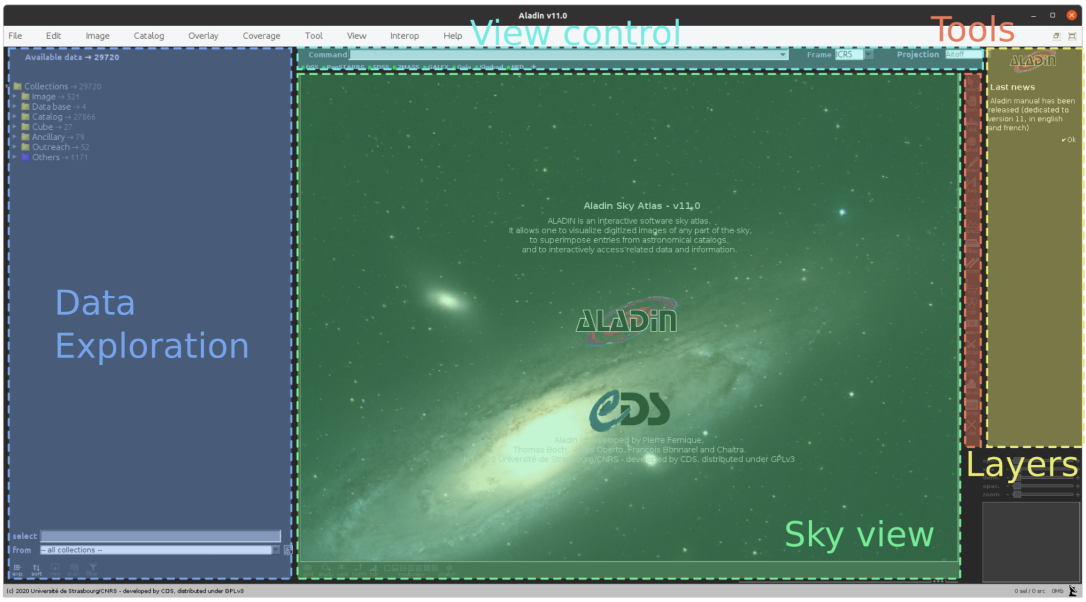
Objectives

The objective of this hands-on session is to introduce some of the most common Virtual Observatory (VO) tools, such as `aladin` (Bonnarel et al., 2000) and `TOPCAT` (Taylor, 2005), and a widely used protocol, `cone-search`, to query data from archives.
We will start with a graphical user interface and then move to using `python` scripts to query Application Programming Interfaces (API) and automatize the repetitive tasks.

1 Finding moving objects in an image (field of view)

This exercise introduces the Sky Body Tracker (`SkyBoT`) tool (Berthier et al., 2006). `SkyBoT` is a specialized `cone-search`: it lists all Solar System Objects (SSOs) present within a given field of view **at a given epoch**. Alike all the services hosted at IMCCE on the VO Solar System Portal (VOSSP[1]), `SkyBoT` requests can be submitted in different ways (`aladin`, HTTP, SOAP), and results can be provided in different formats (VOTable, text, html).

We will use `SkyBoT` to illustrate calls to VO services first with a graphical user interface (GUI), `aladin` (Fig. 1), then with a `python` script (in the `../3.2-Query_APIs/3.2.2-How_to_query_an_API.ipynb` notebook).



Aladin Sky Atlas - v11.0

ALADIN is an interactive software sky atlas.
It allows one to visualize digitized images of any part of the sky,
to superimpose entries from astronomical catalogs,
and to interactively access related data and information.

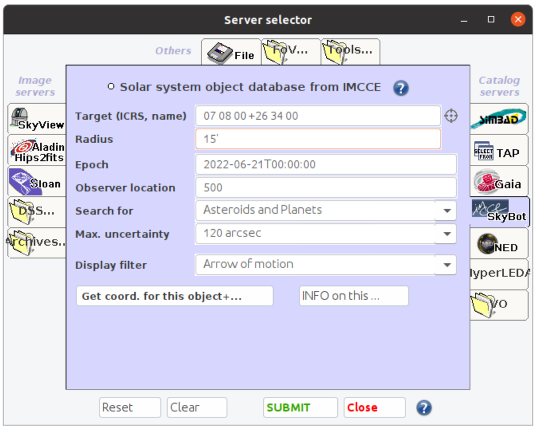
Last news

Aladin manual has been released (dedicated to version 11, in english and french)

Fig. 1: The `aladin` client. Loaded data from the `data exploration` panel are listed as `layers` and displayed in the `sky view`. Coordinates can be entered in the `view control` panel, and many `tools` are available.

[1] Forms: https://ssp.imcce.fr/forms
APIs: https://ssp.imcce.fr/webservices

1. Launch **aladin**.
2. Using the **View Control** (see Fig. 1), center de view on RA = $07^h\,08^m$, DEC = $+26°\,34^m$ [Tips: 1]
3. Using the **Data Exploration**, open the SkyBoT interface [Tips: 2]



Server selector

○ Solar system object database from IMCCE

Target (ICRS, name): 07 08 00 +26 34 00

Radius: 15'

Epoch: 2022-06-21T00:00:00

Observer location: 500

Search for: Asteroids and Planets

Max. uncertainty: 120 arcsec

Display filter: Arrow of motion

There are seven fieds in the pop-up window: **Target**, **Radius**, **Epoch**, **Observer location**, **Search For**, **Max. uncertainty**, and **Display filter**. By default, **aladin** fills the **Target** and **Radius** fields to correspond to current display.

4. Check that the coordinates in Target correspond to the value above.
5. Set the Radius to 15'.
6. Set the Epoch to 2022-06-21T00:00:00.
7. Let everything else by default and click on SUBMIT.

A new layer has appeared, showing the SSOs with their arrow of motion. Like for any other catalog loaded in aladin, you can select objects and see their properties in the bottom panel.

8. Select all SSOs. Ctrl-A or Edit → Select all objects.
9. Flyover any of the column in the bottom panel. An histogram of values for all objects will appear at the bottom right (except for columns Name, RA and DEC).
10. Flyover the Class column then flyover the histogram to see the corresponding targets in the FoV.

We just used a **tiny** fraction of what **aladin** can do. With it, you can retrieve catalogs, images, cross-match them, select intersections, unions, etc. It is an extremely powerful tool to explore the sky.

Tips

[1] enter the string "07 08 +26 34" in the Command field

[2] enter "skybot" in the select field, wait a couple of seconds then select Sky Body Tracker in the list of collections and finally Load. Alternatively, open the Server selector (File → Open server selector, shortcut Ctrl-L) and select SkyBoT in the Catalog servers (right tab).

2 In a script

We often process many images. Clicking on a GUI is not the most productive method. We thus often write scripts to perform many automated steps. Most VO services can be called through APIs and be scripted.

Generally, the service has a base url to which queries are to be sent. The user choices are submitted through a list of parameter=value arguments. Of course, we cannot guess which parameter is available, nor what value are allowed. So read the documentation!

For instance, we can repeat the query above in python (see scripts/query_skybot.py):

```python
import requests
import json
import pandas as pd
from astropy.coordinates import Angle

# User choices
ep = '2022-06-21T00:00:00'
ra = '07h08m00'
dec = '+26d34m00'
sr = 15/60
```

```
observer = ’500’

# Service URL
url = ’https://ssp.imcce.fr/webservices/skybot/api/conesearch.php?’

# Query parameters
params = {
     ’-ep’: ep,
     ’-ra’: Angle(ra).degree,
     ’-dec’: Angle(dec).degree,
     ’-sr’: sr,
     ’-mime’: ’text’,
     ’-output’: ’all’,
     ’-loc’: observer,
     ’-tscale’: ’UTC’
    }

# Query the service
r = requests.post(url, params=params, timeout=2000)

# Write results to disc
with open("response.txt", "w") as f:
    f.write(r.text)

# Load results in a pandas.DataFrame
data = pd.read_csv( ’response.txt’, sep=’|’, skiprows=2 )
```

So long story short, we first define the parameters (coordinates, radius, epoch) then query the `SkyBoT` service thanks to the `request` package. The results are written in a file, and read as a `pandas DataFrame`. Performing this query for each image acquired by a telescope is very easy (the parameters are contained in the image).

The script above is a simple example, let’s move to the notebook `3.3.2-How_to_query_an_API.ipynb` in the directory `3.2-Query_APIs` to get some practice.

References

Berthier, J., Vachier, F., Thuillot, W., et al. 2006, in Astronomical Society of the Pacific Conference Series, Vol. 351, Astronomical Data Analysis Software and Systems XV, ed. C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, 367

Bonnarel, F., Fernique, P., Bienaymé, O., et al. 2000, A&AS, 143, 33

Taylor, M. B. 2005, in Astronomical Society of the Pacific Conference Series, Vol. 347, Astronomical Data Analysis Software and Systems XIV, ed. P. Shopbell, M. Britton, & R. Ebert, 29