Python (Semester 2 2024)

Astronomical Data Sources

Nina Hernitschek
Centro de Astronomía CITEVA
Universidad de Antofagasta

October 14, 2024

Motivation

One of the cornerstones of research is **reproducibility**.

Reproducibility strongly depends on the availability of data.

surveys

data services

Motivation

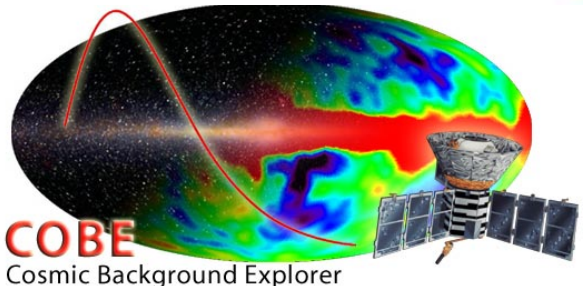
We will see an introduction to **accessing astronomical data with Python**.

In addition, today's session we can also use for answering questions regarding the project (if any).

Surveys as Astronomical Data Sources

Astronomical surveys often share their full data sets. In some cases, these are shared via the observatory that acquired them, for example, the all-sky data acquired with Planck, Wilkinson Microwave Anisotropy Probe, and COBE.

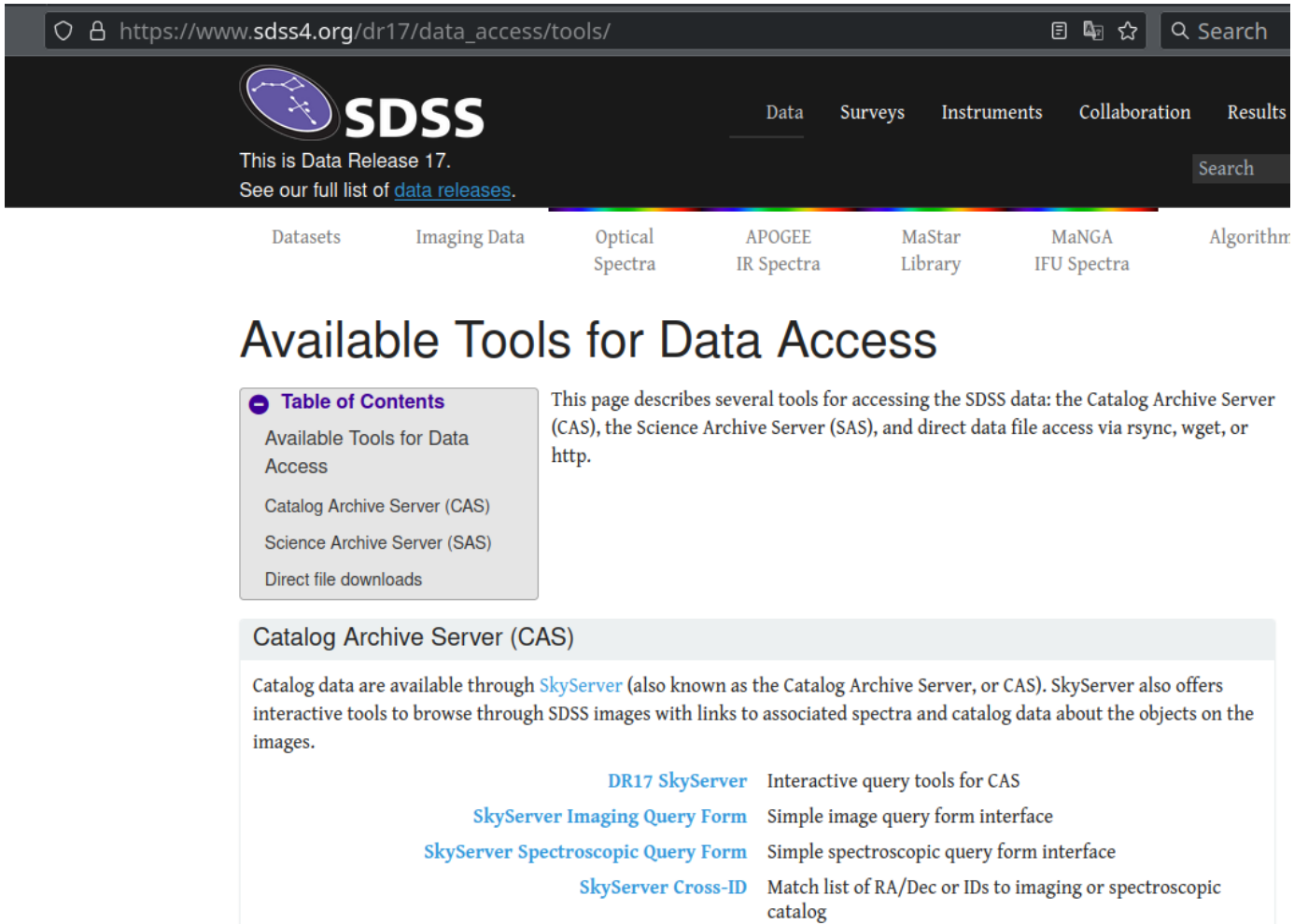
Other surveys serve their own data. Examples include the SDSS, 2MASS, UKIDSS.



Surveys as Astronomical Data Sources

Example: SDSS

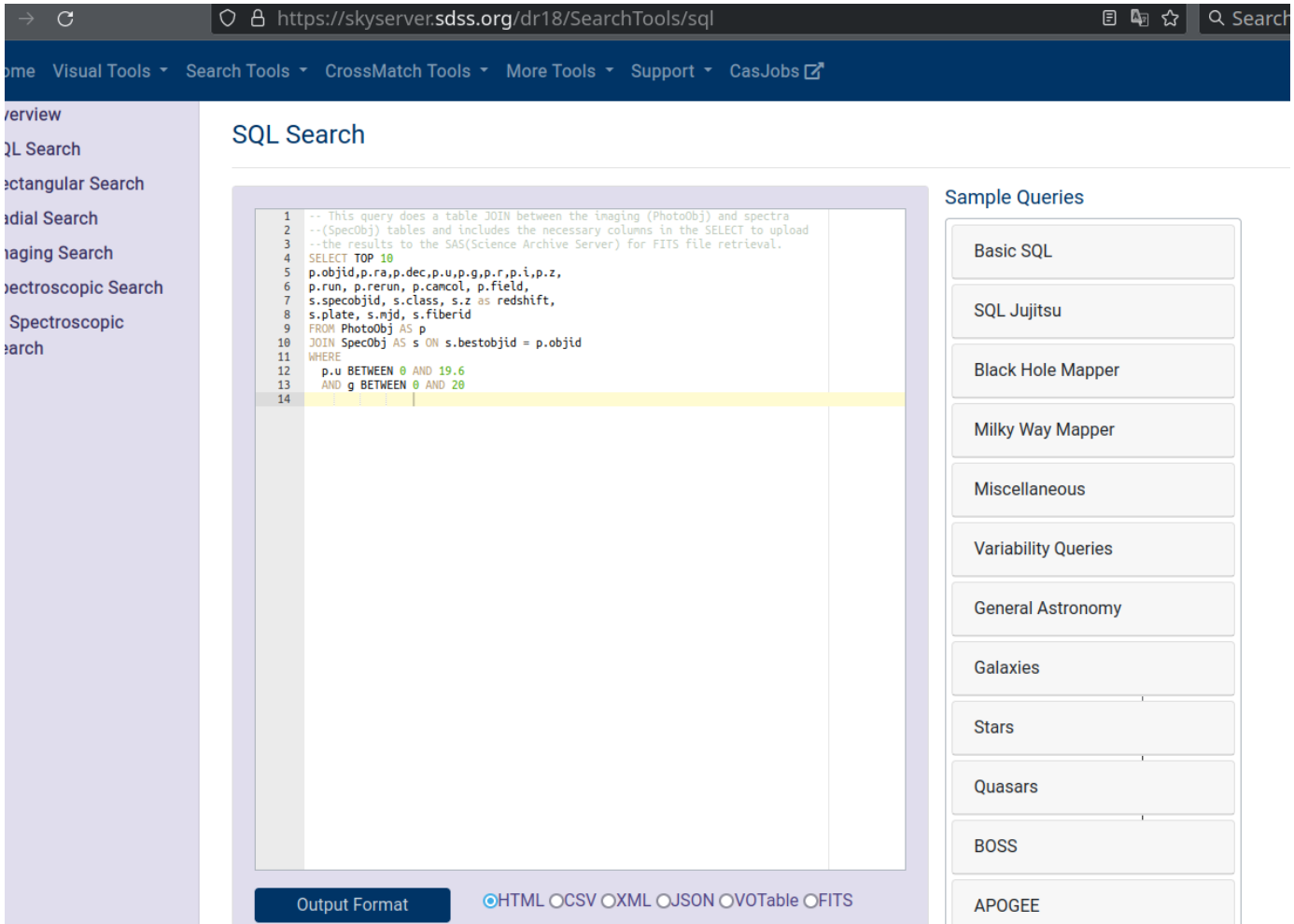
`https://www.sdss4.org`



Surveys as Astronomical Data Sources

Example: SDSS

`https://skyserver.sdss.org`



Surveys as Astronomical Data Sources

Example: SDSS

`sdss-access` provides Python utilities for downloading files from the Science Archive Server (SAS). Examples and more details can be found on the official documentation.

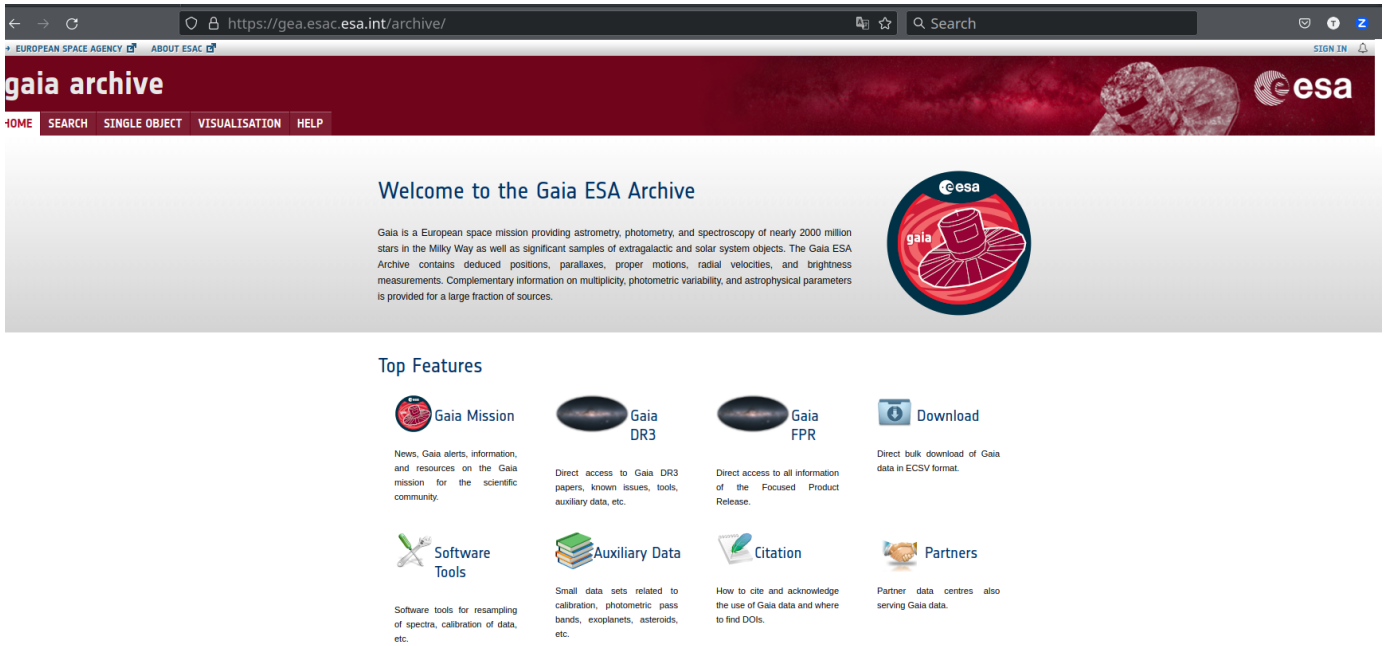
`https://www.sdss.org/dr18/software/packages/sdss-access/`

install `sdss-access` with:

```
pip install sdss-access
```

Surveys as Astronomical Data Sources

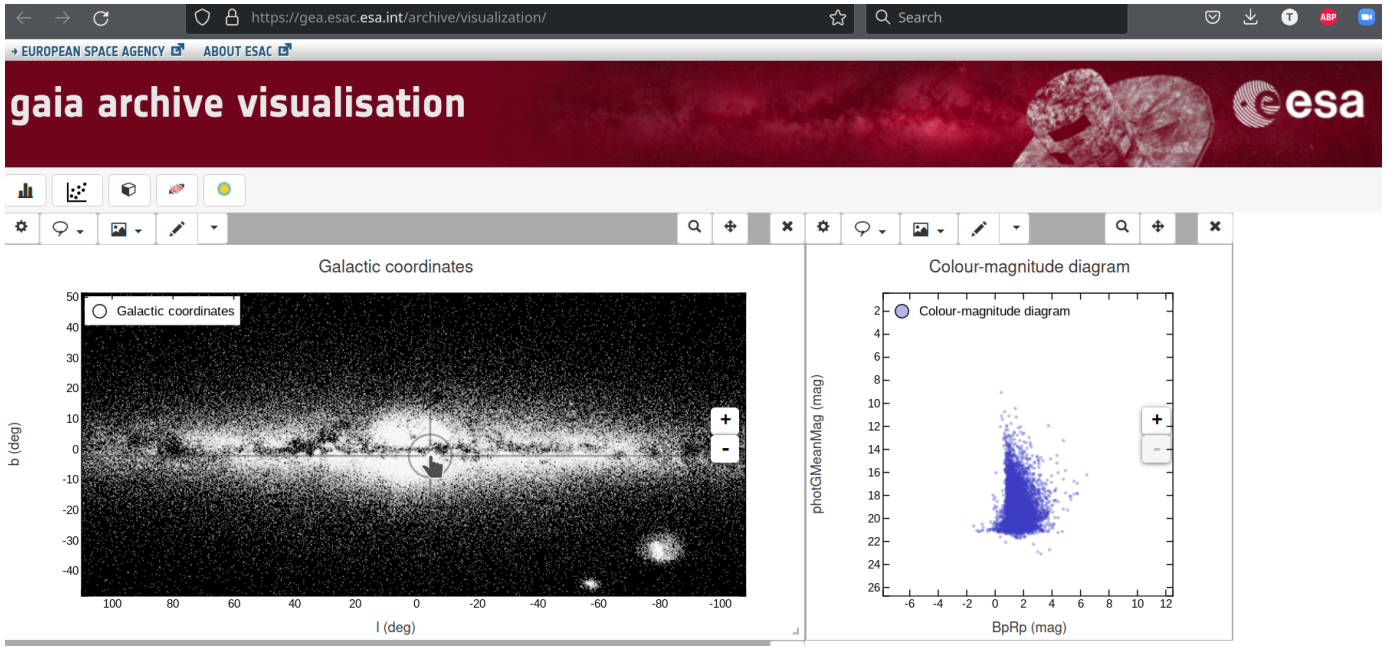
Example: Gaia
`https://gea.esac.esa.int/archive/`



Surveys as Astronomical Data Sources

Example: Gaia

`https://gea.esac.esa.int/archive/visualization`



Surveys as Astronomical Data Sources

Example: Gaia

In addition to the bulk download of data, one can write Astronomical Data Query Language (ADQL) queries which are similar to SQL queries.

example use case: Cone search sorted by angular separation
retrieve a sample of filtered sources (brighter than G=20.5 mag and with a parallax measurement) in a circular region centred on the LMC [(R.A., Dec) = (81.28, -69.78) deg] with a search radius of 5 arcmin. The output should be ordered by the angular separation, from small to large.

Target table: `gaiadr3.gaia_source`

Query:

```
SELECT *, DISTANCE(81.28, -69.78, ra, dec) AS ang_sep
FROM gaiadr3.gaia_source
WHERE DISTANCE(81.28, -69.78, ra, dec) < 5./60.
AND phot_g_mean_mag < 20.5
AND parallax IS NOT NULL
ORDER BY ang_sep ASC
```

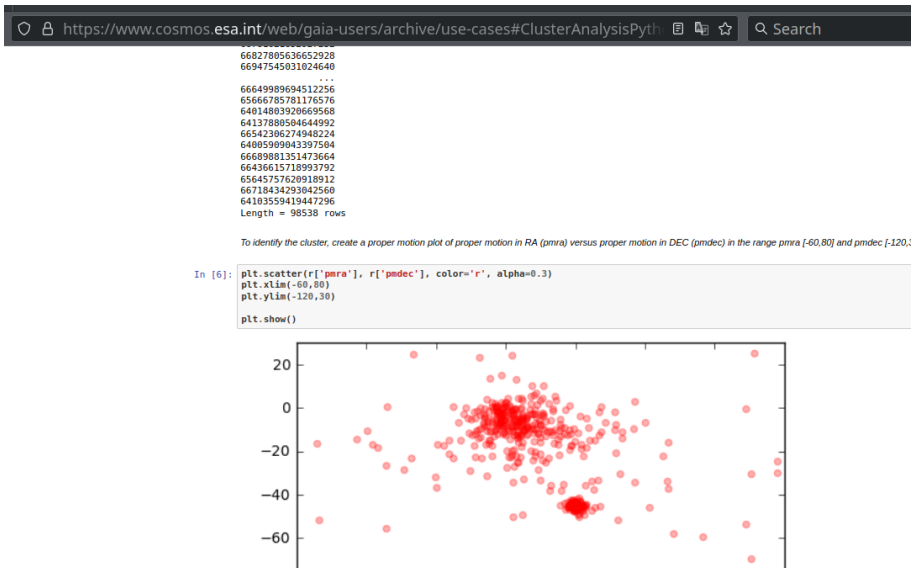
Surveys as Astronomical Data Sources

Example: Gaia

ESA Gaia Archive also provides a Python Astroquery package for easy access to the ESA Gaia Archive: `astroquery.gaia`.

In addition, a generic package is provided for accessing any TAP-compliant service: `astroquery.utils.tap`.

An example Jupyter notebook is provided: `https://www.cosmos.esa.int/web/gaia-users/archive/use-cases#ClusterAnalysisPythonTutorial`



Services

With a lot of different surveys existing, a **common scripted interface** to tie all these services together is a good way to make the data more accessible.

Doing so, also provides **reproducibility** of analysis done for publications. A centrally maintained library also safeguards against lost links to data archives, moving some of the responsibility for maintaining long-term reproducibility from each individual researcher to the broader community.

CDS Services

The Strasbourg astronomical Data Center (CDS) is a data center dedicated to the collection and worldwide distribution of astronomical data and related information. It is located at the Strasbourg Astronomical Observatory, France.

The CDS gives access to many astronomical data such as objects, catalogues, images and spectra.

Several services are available to search for and to get these data. We will look here at two of their main services.

CDS Services

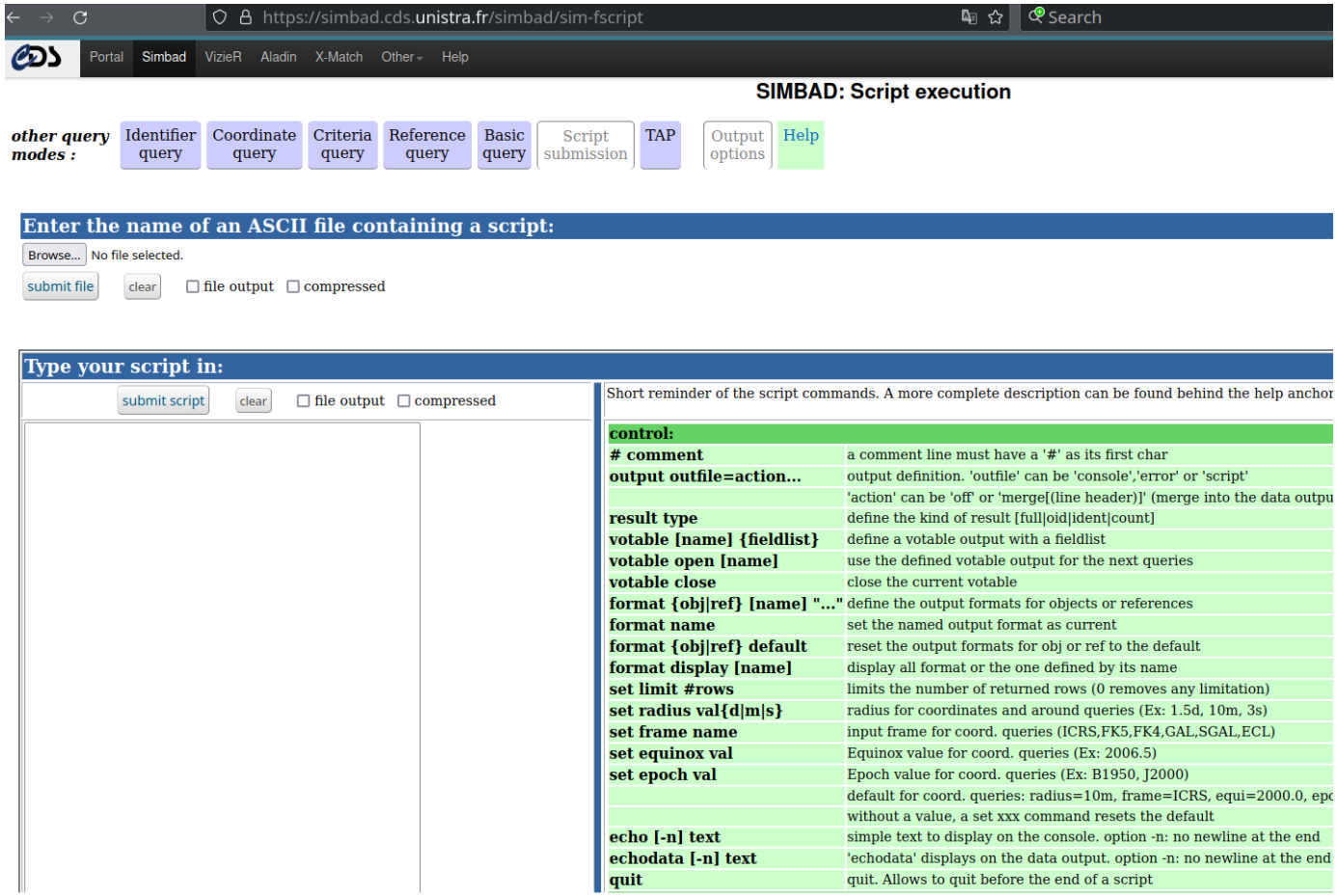
The SIMBAD astronomical database provides basic data, cross-identifications, bibliography and measurements for astronomical objects outside the solar system. SIMBAD can be queried by object name, coordinates and various criteria. Lists of objects and scripts can be submitted. Links to some other on-line services are also provided.

The **SIMBAD** database currently contains information for about 13,000,000 astronomical objects (stars, galaxies, planetary nebulae, clusters, novae and supernovae, etc.).

The only astronomical objects specifically excluded from SIMBAD are Solar System bodies (planets, satellites, asteroids, comets)

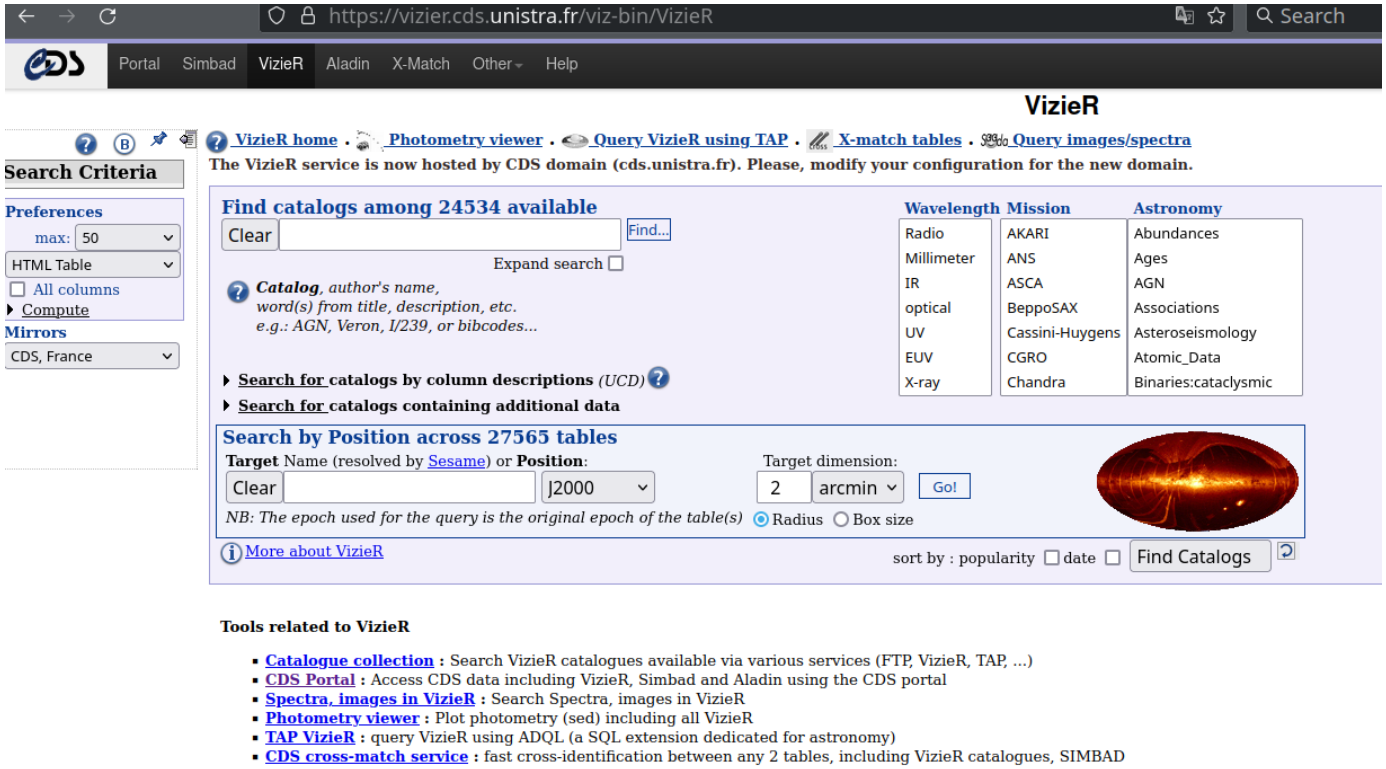
CDS Services

`https://simbad.unistra.fr/simbad/`



CDS Services

The **VizieR** service provides mostly catalogs associated with papers or authors.
`https://vizier.cds.unistra.fr/viz-bin/VizieR`



CDS Services

An extensive list of Python tutorials on how to interface with the CDS services is available:

`https://cds-astro.github.io/tutorials/intro.html`

Community brokers

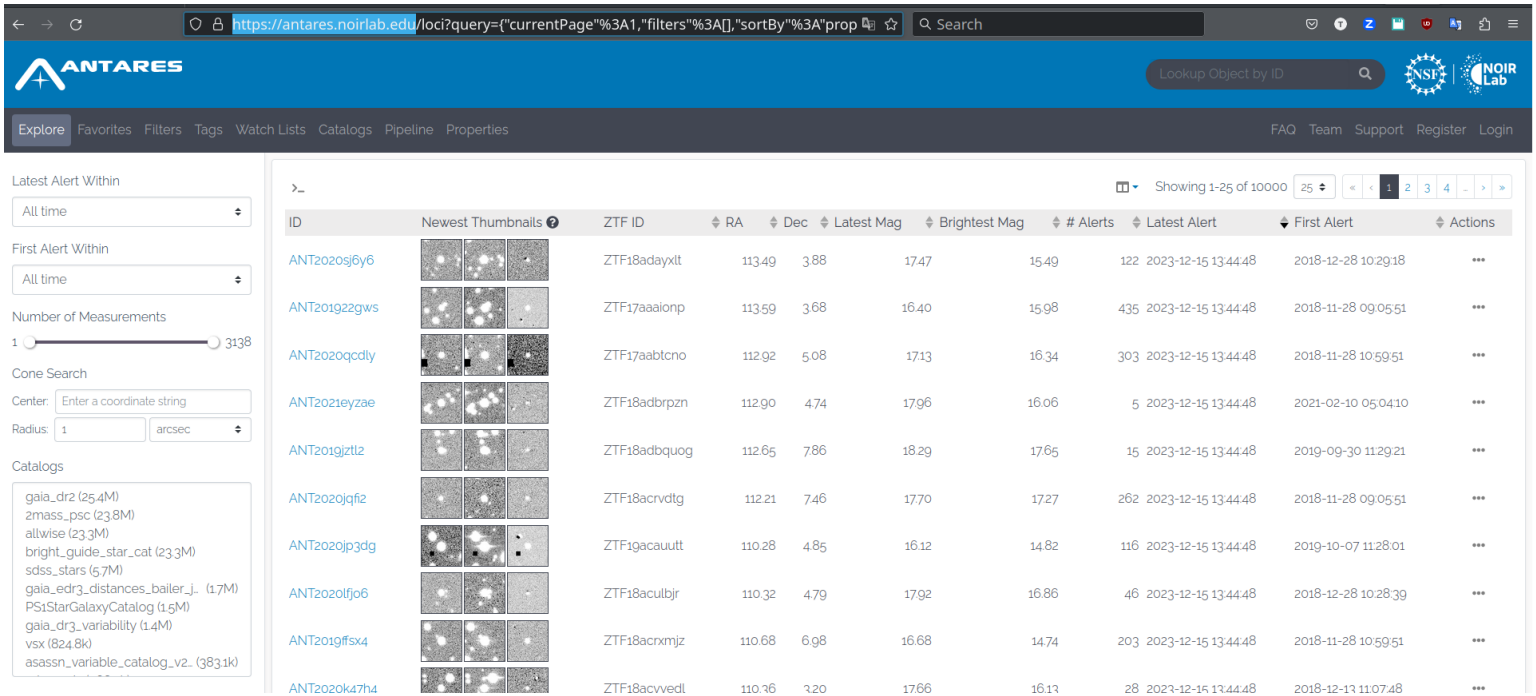
The newest generation of large survey telescopes such as the Vera C. Rubin Observatory (which will conduct the 10-year Legacy Survey of Space and Time (LSST)) are designed to produce large streams of transient alerts. They will be processed, annotated and classified by a new generation of astronomical software for time-domain surveys, called **alert brokers**.

Brokers are currently developed tested with the Zwicky Transient Facility (ZTF). Usually, brokers provide a website as well as a Python interface.

Community brokers

Example: The ANTARES Broker

`https://antares.noirlab.edu/`

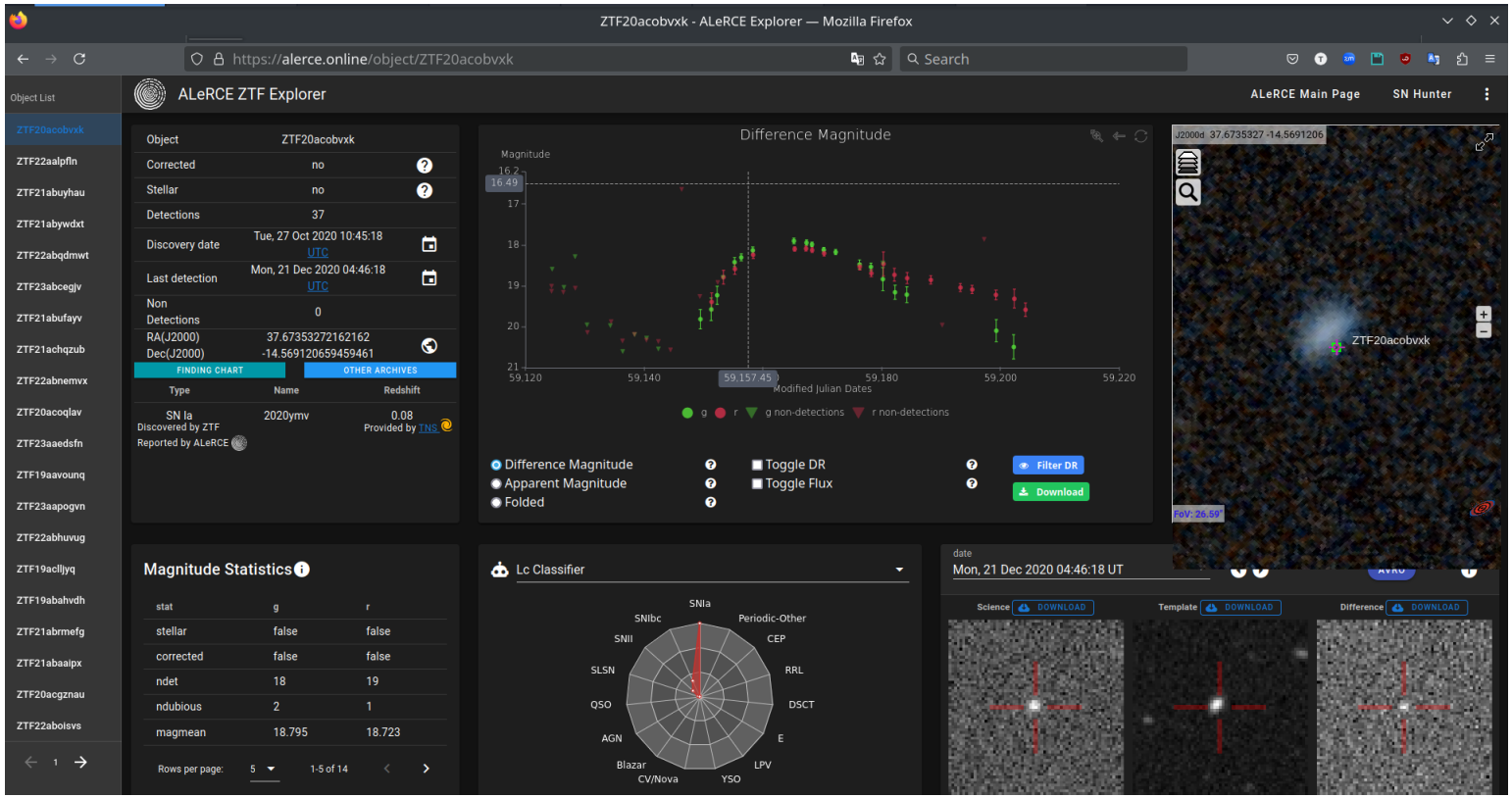


Community brokers

Example: The ALeRCE Broker

`http://alerce.science/`

`https://github.com/alercebroker/usecases`



Community brokers

Brokers will show data from LSST as soon as LSST will deliver the alert stream.

This will be from approx. mid 2025 on.

What is astroquery?

Astroquery as part of astropy is a set of tools for querying astronomical web forms and databases.

In addition to providing access to astronomical databases, astroquery provides data such as molecular and atomic properties, e.g. by the NIST Atomic Spectra Database, bibliographic databases such as the NASA Astrophysics Data System (ADS), or services that are computationally intensive or require regular updates, like solar system ephemerides provided by services like JPL HORIZONS, or the Minor Planet Center.

https://github.com/astropy/astroquery
https://astroquery.readthedocs.io/

What is astroquery?

These tools are built on the Python `requests` package, which is used to make HTTP requests, and `astropy`, which provides most of the data parsing functionality. astroquery modules generally attempt to replicate the web page interface provided by a given service as closely as possible, making the transition from browser-based to command-line interaction easy.

The complete documentation for `astroquery` can be found at

`http://astroquery.readthedocs.io/`.

Using astroquery

All astroquery modules are supposed to follow the same API. In its simplest form, the API involves queries based on coordinates or object names. Some simple examples, using SIMBAD:

```
from astroquery.simbad import Simbad

result_table = Simbad.query_object("m1")

print(result_table)
```

```
MAIN_ID     RA         DEC     ...    COO_BIBCODE      SCRIPT_NUMBER_ID
          "h:m:s"    "d:m:s"   ...
------- ---------- --------- ... ------------------- ----------------
  M   1 05 34 30.9 +22 00 53 ... 1995AuJPh..48..143S                1
```

Using astroquery

All query tools allow coordinate-based queries:

```
from astropy import coordinates
import astropy.units as u

# works only for ICRS coordinates:

c = coordinates.SkyCoord("05h35m17.3s -05d23m28s", frame='icrs')
r = 5 * u.arcminute

result_table = Simbad.query_region(c, radius=r)
print(result_table)
```

```
        MAIN_ID                RA       ...     COO_BIBCODE     SCRIPT_NU
                             "h:m:s"     ...
----------------------- ------------- ... ------------------- ---------
        NAME Ori Region   05 35 17.30 ...
                   ...            ... ...                 ...
2MASS J05353573-0525256 05 35 35.7755 ... 2020yCat.1350....0G
           V* V2114 Ori 05 35 01.6720 ... 2020yCat.1350....0G
Length = 3272 rows
```

Using astroquery

The following is an alphabetic **selection of available services** through astroquery; the full list can befound in the documentation:

ALMA Queries (astroquery.alma)
Atomic Line List (astroquery.atomic)
Besancon Queries (astroquery.besancon)
CADC (astroquery.cadc)
CASDA Queries (astroquery.casda)
Cologne Database for Molecular Spectroscopy (CDMS) Queries (astroquery.linelists.cdms)
DACE (astroquery.dace)
ESA Herschel Science Archive (astroquery.esa.hsa)
ESA HST Archive (astroquery.esa.hubble)
ESA ISO Archive (astroquery.esa.iso)
ESA JWST Archive (astroquery.esa.jwst)
ESA XMM-Newton Archive (astroquery.esa.xmm_newton)

Chosing the Right Data Source

Which **survey** you use is mostly determined by the science you are doing.

How you **access** the data is mostly determined by the amount of data.
Often, a survey provides multiple ways for accessing the same data.

Chosing the Right Data Source

Small data sets, exploratory:
Explore them directly on the website, e.g. clicking through images, plots.
example:

- Using the broker website.
- Using small, older surveys with digitized data.

Chosing the Right Data Source

Small data sets, exploratory:
Explore them directly on the website, e.g. clicking through images, plots.
example:

- Using the broker website.
- Using small, older surveys with digitized data.

Medium-sized data sets:
Download survey data ("bulk download") and process locally.
example:

- Downloading a catalog of variable stars in Pan-STARRS.
- Downloading a sector of TESS data to search for specific objects.

Chosing the Right Data Source

Small data sets, exploratory:
Explore them directly on the website, e.g. clicking through images, plots.
example:

- Using the broker website.
- Using small, older surveys with digitized data.

Medium-sized data sets:
Download survey data ("bulk download") and process locally.
example:

- Downloading a catalog of variable stars in Pan-STARRS.
- Downloading a sector of TESS data to search for specific objects.

Very large data sets:
Access data using Python packages. Select data using e.g. SQL/ADQL, download only selected data. Process locally or in the cloud.
example:

- Accessing SDSS, Gaia.
- Python broker interfaces, downloading only specific alerts.

Summary & Outlook

We have seen ways on how to access astronomical databases from within Python.

In the next lecture we will continue with **Astronomical Packages** which provide us with functionality such as coordinate conversation, data reduction (photometry, spectroscopy) and other more specialized calculations in astronomy.