# Astroquery.Simbad: Cataloging… Please Wait…

May 18, 2025

Jeremy Vo

1. Astroquery is a package that provides python with the ability to query astronomical web-based astronomy archives and catalogues. This package is used to efficiently call data from web based astronomy services, instead of having to manually download data through their, often slow, websites. There are currently 45 available modules that Astroquery supports. Catalogues such as SIMBAD, VizieR, or even ESASky can be accessed from your local IDE. Archives such as IPAC IRSA, are also available to retrieve FITS images or spectra data.

2. As an Astronomy major, I chose Astroquery due to the relevance it has in observational analysis. Last semester, I utilized astroquery.simbad, as well as astroquery.ipac.irsa, to investigate the possibility of using TRGBs (Tip of the Red Giant Branch stars) as a form of standard candle. I utilized distance measurement queries from SIMBAD and magnitude data from IRSA all via astroquery. Due to the span of Astroquery, I decided to focus on Astroquery.Simbad. SIMBAD is one of the more centralized databases that has over 11 million objects that have been cross-referenced by other databases and sources. SIMBAD allows astronomers access to quick, reliable information about these objects. However, their web-based search is absolute crap, being extremely slow for searches with over 500 objects. Astroquery.simbad can address this problem by bypassing SIMBAD's UI.

3. Astroquery was first copyrighted in 2011[1], with its first public version launching in 2013[2]. Its foundation is built on the 'requests' package, which allows users to make HTTP web queries. Astropy is also necessary, as it provides astroquery its critical data parsing capabilities.[3] The Astroquery website says that pyvo and Simple-Cone-Search-Creator offer complimentary functions. Even the Space Telescope Science Institute recommends utilizing astroquery (astroquery.esa.hubble) for data.[4] I've installed astroquery version 0.4.10, based on the version checker.

---

[1] https://astroquery.readthedocs.io/en/latest/license.html
[2] https://github.com/astropy/astroquery/releases?page=3
[3] Ginsburg, A., Sipőcz, B., Brasseur, C., Cowperthwaite, P., Craig, M., Deil, C., Guillochon, J., Guzman, G., Liedtke, S., Lian Lim, P., Lockhart, K., Mommert, M., Morris, B., Norman, H., Parikh, M., Persson, M., Robitaille, T., Segovia, J.C., Singer, L., Tollerud, E., de Val-Borro, M., Valtchanov, I., Woillez, J., Astroquery Collaboration, & a subset of astropy Collaboration (2019). astroquery: An Astronomical Web-querying Package in Python. \aj, 157(3), 98.
[4] https://hst-docs.stsci.edu/hstdhb/4-hst-data-analysis/4-3-notable-python-packages-for-hst

4. The package is still being maintained by the original authors (Adam Ginsburg & Brigitta M. Sipőcz).[5] The package had 77 contributors in 2018, with some of them being directly affiliated with provided module services (ESA Gaia team, ADS team, etc).[6] According to them, “Anyone can contribute to astroquery” and they welcome changes from “both individuals and institutions”[7]. Contributions to astroquery have to follow a specific template module and by opening a pull request with the maintenance team; they also recommend following Astropy contribution guidelines (as a coordinated package of Astropy).[8][9]

5. Astroquery is extremely easy to install. I used `!pip install astroquery` , and I was practically done. I did use `!pip install --upgrade astroquery astropy` to verify I had the latest version.

6. Yes, astroquery installed with the “standard” pip. See question 5 for installation.
7. The source code is available by utilizing the `print(astroquery.simbad.__file__)` function to find the `__init__.py` file. You can also find the source code on github here: https://github.com/astropy/astroquery/blob/main/astroquery/simbad/__init__.py

8. I have not seen any packages that necessarily need astroquery to function correctly.

9. Astroquery.simbad can be used in a python script or a jupyter notebook.

10. In my project, I make heavy use of the query_region function. Here’s an example searching for red giant branch stars in a 2 arcmin cone, centered at the north pole (using astropy.coordinates.SkyCoord and astropy.units):
```
radius = 1*u.deg
pole = SkyCoord(0, 90, unit=("deg", "deg"))
Rgbstars = simbad.query_region(pole, radius=radius, criteria = “otype=’RGB*’”)
```

11. The astroquery.simbad does not produce figures. I needed to utilize matplotlib for my figure.

---

[5] Ginsburg, A., Sipőcz, B., Brasseur, C., Cowperthwaite, P., Craig, M., Deil, C., Guillochon, J., Guzman, G., Liedtke, S., Lian Lim, P., Lockhart, K., Mommert, M., Morris, B., Norman, H., Parikh, M., Persson, M., Robitaille, T., Segovia, J.C., Singer, L., Tollerud, E., de Val-Borro, M., Valtchanov, I., Woillez, J., Astroquery Collaboration, & a subset of astropy Collaboration (2019). astroquery: An Astronomical Web-querying Package in Python. \aj, 157(3), 98.

[6] Ibid
[7] Ibid
[8] https://github.com/astropy/astroquery/blob/main/CONTRIBUTING.rst
[9] https://www.astropy.org/contribute.html

12. I mapped globular clusters with distances less than 30 kpc from Earth. See the bottom for the image and caption. My notebook shows the queries utilized to obtain the coordinate data.

13. Astroquery and its service submodules are all in pure python. Its dependencies, such as Astropy and requests, are both written in pure python.

14. Astroquery.simbad requires the user to fill in certain parameters, which is somewhat dependent on your query mode. For example, `simbad.query_object()` requires an object name, whereas `simbad.query_region()` requires right ascension, declination, radius of the search, and object criteria. Object criteria syntax is listed in SIMBAD's Object types page: https://simbad.cds.unistra.fr/guide/otypes.htx (In my case, globular clusters is 'GlC')

15. The output of astropy.simbad consists of columns of data (float or string). Basic information includes main_id (name), ra (right ascension), dec (declination), and coordinate error. Additional columns can be added to the output using `simbad.add_votable_fields` (in my case, ra, dec, and object distance being the most critical fields). This can be stored normally in a numpy array.

16. There are tests that prod web connection and remote data collection capabilities in order to ensure you're querying a service submodule correctly. You can find these tests in astroquery/simbad/tests.[10]

17. As an alternative for double checking and running the tests, you can use SIMBAD's own web-based search with the same test criteria. You can then double check your results from both the web table and your python dataset.

18. Astropy and results packages are necessary for Astroquery to function properly. I found this in a paper the developers provided for citation.[11]This makes sense; I was using coordinates synthesized by astropy in order to use the query_region function. Without results, astroquery wouldn't even be able to connect to the web.

19. Astroquery.simbad has a comprehensive guide on their dedicated document website: https://astroquery.readthedocs.io/en/latest/simbad/simbad.html.
    It provides examples for all its functions. If that's not enough, you can also look at SIMBAD's own syntax guide present on here:
    https://simbad.u-strasbg.fr/simbad/sim-fsam

---

[10] https://github.com/astropy/astroquery/tree/main/astroquery/simbad/tests

[11] Ginsburg, A., Sipőcz, B., Brasseur, C., Cowperthwaite, P., Craig, M., Deil, C., Guillochon, J., Guzman, G., Liedtke, S., Lian Lim, P., Lockhart, K., Mommert, M., Morris, B., Norman, H., Parikh, M., Persson, M., Robitaille, T., Segovia, J.C., Singer, L., Tollerud, E., de Val-Borro, M., Valtchanov, I., Woillez, J., Astroquery Collaboration, & a subset of astropy Collaboration (2019). astroquery: An Astronomical Web-querying Package in Python. \aj, 157(3), 98.

This was more than enough for me to get the hang of my queries.

20. Astroquery asks that you just cite the paper Ginsburg, Sipőcz, Brasseur et al 2019 whenever you use astroquery. You just need to retrieve its BibTex and you're all good.

21. See footnotes for other sources.

22. Two papers that utilizes Astroquery:
    - Paletou, F., Bohm, T., Watson, V., & Trouilhet, J.F. (2015). Inversion of stellar fundamental parameters from ESPaDOnS and Narval high-resolution spectra. åp, 573, A67.
    - Vogt, F., Mehner, A., Figueira, P., Yu, S., Kerber, F., Pfrommer, T., Hackenberg, W., & Bonaccini Calia, D. (2023). Pure-rotational and rotational-vibrational Raman spectrum of the atmosphere at an altitude of 23 km. Phys. Rev. Res., 5, 023145.

23. No new python methods were necessary to learn for my project goals. I simply utilized an if statement to comb through empty rows, created numpy arrays with coordinate data, and did some simple coordinate transformations to plot it in matplotlib.pyplot.

24. Disclaimer: I have prior experience using astroquery.simbad for ASTR 310.

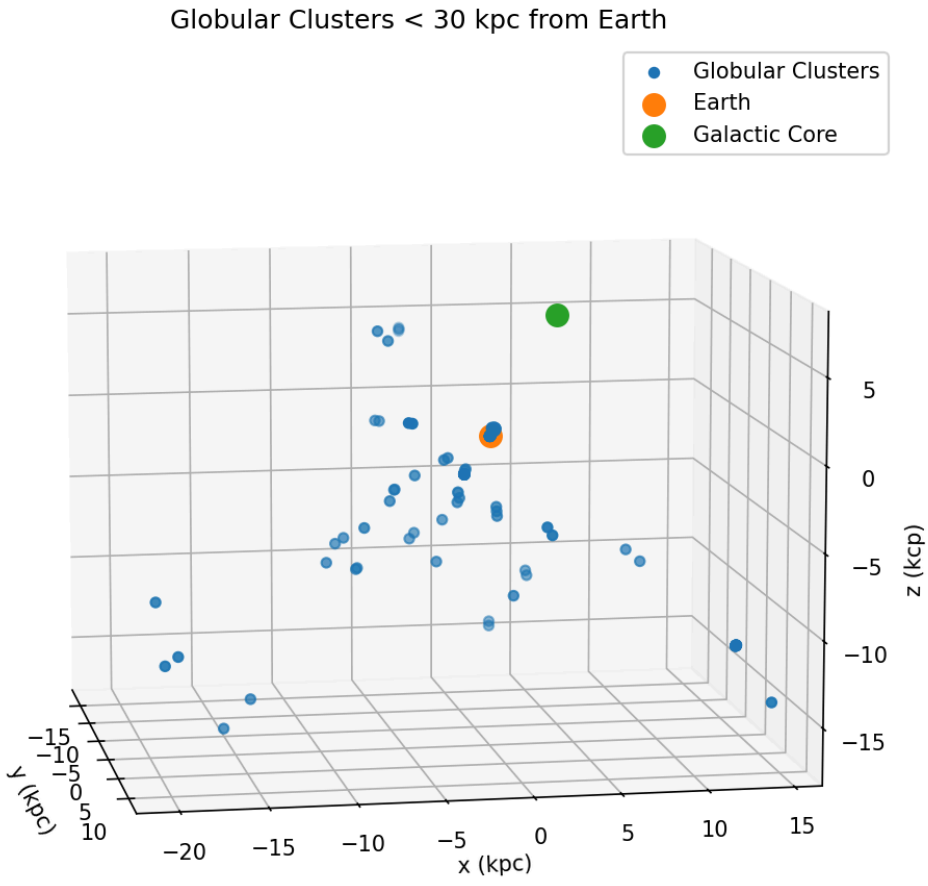

**Figure 1:** A 3d plot of globular clusters within 30 kiloparsecs of Earth. Along with Earth, I've plotted where the galactic core should be (around Sagittarius A*). As one can see, we can see many more globular clusters outwards, upwards, and downwards, rather than corewards. This makes sense, as the coreward observation is inhibited by tons of gas and interstellar matter.