The Astroquery Package and the Astroquery.Simbad Module

1. Astroquery.Simbad: Reaching for the Stars: A Neighborhood Tour Within 10 Parsecs of the Sun

   Edward Bertrand - May 18th, 2025

2. Astroquery.simbad is a module of the astroquery Python package that provides a convenient interface for accessing the SIMBAD astronomical database directly from Python. With astroquery.simbad, users can query SIMBAD for information about astronomical objects, acquire their coordinates, classifications, and references all within Python scripts or interactive sessions, which is much more convenient than downloading the data from SIMBAD and putting it into a script. Overall, the astroquery package can handle many databases including CosmoSim, Exoplanet Orbit, HITRAN, LAMDA, NED, NIST Atomic Spectra, Splatalogue, VAMDC, Simple Cone Search, VizieR, NASA Exoplanet Archive, NASA ADS, SkyView, MAST, SDSS, Gaia Archive, ESO Archive, JPL Horizons, MPC, IRSA, ALMA Science Archive, and UKIDSS.

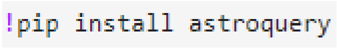
3. I opted to use the astroquery.simbad module because, being an astronomy student, I have to download various astronomical datasets from various sources frequently (specifically SIMBAD). Before, I downloaded data from these sources separately and then imported them into Python for analysis, which was really tedious and time-consuming. Astroquery.simbad does all this processing for me by allowing me to acquire this data directly from SIMBAD in my Python script. This is not only time-saving, but it also makes the data acquisition reproducible and automated, which is essential to coursework and astronomical research.

4. Astroquery.simbad was originally released in September of 2013 by Adam Ginsburg, with the first beta release of the astroquery package and as of today, this package has had contributions from 192 individuals. However, astroquery.simbad has had contributions from 28 individuals, which was found by running the following command in the terminal within the cloned astroquery repository: git log --pretty=”%an” – astroquery/simbad | sort | uniq -c | sort -nr, which lists all the contributors who have made commits in the astroquery/simbad folder. Astroquery.simbad follows the evolutions of the SIMBAD database which is very convenient. Before astroquery.simbad, the SIMBAD database was accessed directly using web forms, custom scripts, or downloading data manually. After astroquery.simbad was developed, other modules in astroquery were developed to access different astronomical databases such as VizieR. There are two other packages that have similar features to astroquery, PyVO, which is an astropy affiliated package, and Simple-Cone-Search-Creator to generate a search service. These are more oriented to

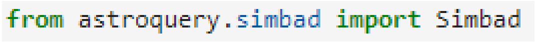
general virtual observatory discoveries and queries, while astroquery has web service specific interfaces. The version of astroquery.simbad corresponds to the current version of astroquery since it is a module of the package, and the current version is 0.4.10.

5. The primary maintenance of the astroquery package is typically handled by two individuals at a time, usually Adam Ginsburg and Brigitta M. Sipőcz, however most individual modules have been implemented independently by contributors. The astroquery.simbad module is handled by Adam Ginsburg, and Frédéric Grollier. There are instructions on how to contribute to the astroquery package on astroquery's GitHub at https://github.com/astropy/astroquery/blob/main/CONTRIBUTING.rst.

6. Installing this package was very simple and quick, only taking a few seconds on my MacBook. The installation was straightforward using the command seen below. Importing the astroquery.simbad module was very simple as well as seen by the import statement below the command installing the astroquery package. After this, SIMBAD queries can be performed directly within Python scripts.



```
!pip install astroquery
```



```
from astroquery.simbad import Simbad
```

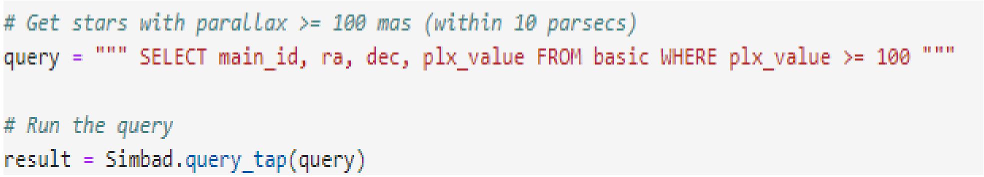
7. This package installs via the "standard" pip/conda commands and the module imports via the standard import statement for other modules such as numpy, making this process straightforward and familiar for most Python users.

8. The source code for astroquery.simbad is available on GitHub as part of the astroquery repository at https://astroquery.readthedocs.io/en/latest/_modules/astroquery/simbad.html, and the source code for astroquery.simbad.core is found at https://astroquery.readthedocs.io/en/latest/_modules/astroquery/simbad/core.html. This allows you to inspect the implementation, download the code, and modify it as needed. Installing astroquery via pip gives you the package, but cloning from GitHub lets you access and explore the full source code directly.

9. After extensive research, I could not find specific examples of astroquery.simbad or astroquery being used in specific packages, but I was able to find a scientific paper and project that uses the astroquery.simbad module and the astroquery package (Sources found in section 22).

10. The code is intended for use within a Python environment, either as a standalone script or interactively in a Jupyter notebook, not as a command line tool or web application. This

can be seen in the attached Jupyter notebook. Below is an example of how the code is used to query objects.



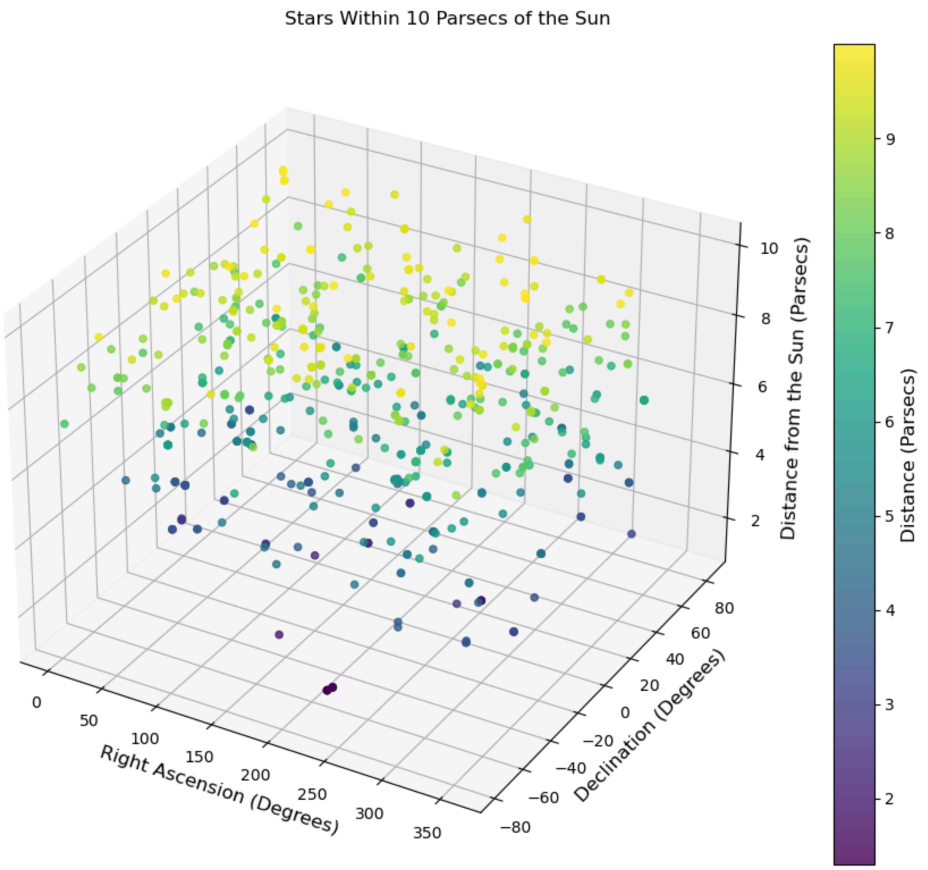
```
# Get stars with parallax >= 100 mas (within 10 parsecs)
query = """ SELECT main_id, ra, dec, plx_value FROM basic WHERE plx_value >= 100 """

# Run the query
result = Simbad.query_tap(query)
```

11. The accompanying Jupyter notebook depicts how I have used the astroquery.simbad module and demonstrates what can be done with the queried data.

12. The astroquery.simbad module does not produce figures, instead it returns the results for the query in a table. A simple way to plot the results of the query is using matplotlib’s plotting capabilities. Visualizing the results is left to the user, who can choose any preferred method.

13. Below is a 3D scatter plot showing stars that are within 10 parsecs of the Sun, plotted by their right ascension, declination, and distance. The colorbar corresponds to each star's distance from the Sun. Note that there are many overlapping stars due to them sharing the same right ascension and declination values (coordinate values).



14. Astroquery.simbad is written entirely in Python, and its design allows users to access SIMBAD directly from Python scripts or interactive environments. Most work is done using Python commands, meaning there is no need for shell scripting or C/C++/Fortran code, unlike other packages.

15. The input for astroquery.simbad is a set of parameters or a query string that specifies what type of data to retrieve from the SIMBAD database. In the provided Jupyter notebook you can see an example of the input being a query that selects stars with parallax greater than or equal to 100 milliarcseconds, which selects stars within 10 parsecs of the Sun. Astroquery.simbad does not require you to provide a dataset or file as an input, data is acquired from the SIMBAD database. There are no built-in tools in astroquery.simbad to generate synthetic data from scratch since it is designed to query SIMBAD which contains existing astronomical data.

16. The output of astroquery.simbad is datasets in the form of tables containing the queried astronomical data, which can be further processed, analyzed, or visualized in any method

that is preferred by the user. In the attached Jupyter notebook I decided to put the queried data into a scatter plot and pandas dataframe.

17. Astroquery.simbad does provide unit tests. There are test files named “test_simbad.py” , “test_simbad_remote.py”, and “test_utils.py” in the tests folder on GitHub, which implies that the code includes testing for functionality and remote queries, which ensures correctness and to catch regressions. As of right now, I do not see any mention of benchmarking tests.

18. I am confident of my results being reliable from these unit and regression tests, which proves that the code is performing the way it should and that any modifications do not introduce any issues. Running these tests regularly ensures that any faults are detected and that these results are repeatable. Additionally, astroquery.simbad relies on information from the SIMBAD database, which is an excellent source that is used by astronomers, and the code itself is updated and verified by the community, ensuring its reliability.

19. Astroquery.simbad mainly depends on pure Python packages such as requests, numpy, astropy, pyVO, keyring, Beautiful Soup, html5lib for installation and use. For running tests, astroquery.simbad depends on curl, pytest-astropy, and pytest-rerunfailures. This information was found on astroquery’s website.

20. The documentation provided was sufficient in guiding me to use astroquery.simbad properly. The documentation can be found at https://astroquery.readthedocs.io/en/latest/simbad/simbad.html and https://astroquery.readthedocs.io/en/latest/index.html. These websites allowed me to fully discover how to properly use astroquery and how to query the SIMBAD database, which was initially confusing at first because I attempted to query in the same way that you would query on SIMBAD’s official website, but it is not that simple. The website provides detailed API documentation, examples, and tutorials on tasks and complicated queries. The documentation is nicely structured and concise, making it relatively easy to understand how to interact with the SIMBAD database. Overall, the documentation was thorough and sufficient for beginners and advanced users.

21. If this code is used in a paper, the preferred citation is the paper:
    Ginsburg, A., “astroquery: An Astronomical Web-querying Package in Python”, *The Astronomical Journal*, vol. 157, no. 3, Art. no. 98, IOP, 2019. doi:10.3847/1538-3881/aafc33.

    Link to this paper: https://ui.adsabs.harvard.edu/abs/2019AJ....157...98G/abstract

22. References:

    Astroquery preferred citation:
    https://ui.adsabs.harvard.edu/abs/2019AJ....157...98G/abstract

    Astroquery Modules: https://astroquery.readthedocs.io/en/latest/_modules/index.html

    Astroquery GitHub: https://github.com/astropy/astroquery

    Astroquery Read the Docs: https://astroquery.readthedocs.io/en/latest/index.html

    SIMBAD Queries: https://astroquery.readthedocs.io/en/latest/simbad/simbad.html

    Astroquery.simbad source code:
    https://astroquery.readthedocs.io/en/latest/_modules/astroquery/simbad.html

    Astroquery.simbad.core source code:
    https://astroquery.readthedocs.io/en/latest/_modules/astroquery/simbad/core.html

    Maintainers WikiPage: https://github.com/astropy/astroquery/wiki/Maintainers

    Scientific Project including astroquery: Szoke, M.-A. (2024) *Chemical composition detector of stars using spectroscopy*, GitHub. Available at: https://github.com/SzokeMark-Andor/Chemical-composition-detector-of-Stars-using-spectroscopy/tree/main.

    Scientific Paper including astroquery.simbad:
    https://ui.adsabs.harvard.edu/abs/2014arXiv1408.7026P/abstract

23. Using ADS there were 630 papers found in the ASCL reference that use astroquery and these are found at: https://ui.adsabs.harvard.edu/abs/2019AJ....157...98G/citations. When it comes to using the astroquery.simbad module specifically, I only found one paper that mentions that module and it is found at: https://ui.adsabs.harvard.edu/abs/2014arXiv1408.7026P/abstract and this paper was cited 15 times which can be seen here: https://ui.adsabs.harvard.edu/abs/2014arXiv1408.7026P/citations.

24. I did not have to learn many new Python methods to use astroquery.simbad as this course provided me with a solid foundation for what I needed to do once I acquired the data. To

acquire the data I needed to learn how to properly query SIMBAD in Python rather than the actual database itself, however this did not take long as there was a tutorial on how to do this on astroquery's website. I did have to learn additional pandas methods for data manipulation, though, like placing my query results in a dataframe, sorting data, and changing data types. I also learned how to specify column width in Python's f-strings to correctly form the table output, which served to make the results more readable.

25. I have had previous experience using data acquired from SIMBAD in ASTR121 and ASTR310, but I have not had any experience using astroquery or astroquery.simbad before this project, but I will use this package and module in the future to make data analysis easier and more efficient.