Lab 0: Introduction to ADQL and Gaia Data

Astronomy Data Lab AY128 (UC Berkeley)

Assigned: Fri Aug. 22, 2025
Checkpoints:Wed Sep. 3, 2025; Wed Sep. 10, 2025
Final Lab Submission Due: 11:59 PM Sun Sep. 14, 2025

1. Introduction & Logistics for Your Lab Submission

Welcome to the **Astronomy Data Lab**! This first (zero-indexed) lab introduces *Gaia* data and ADQL—a scripting language used to query databases. Lab 0 is shorter than Labs 1–3, but it provides an idea of the workload in Astro 128/256. It will also be useful preparation for Lab 1. The expectations for undergrads and grad students are the same for Lab 0.

For each checkpoint, turn in a PDF rendering of a Jupyter notebook[1] including discussion, code, and the ADQL queries needed to reproduce your results. Explain in words what your code and queries do, including how you made important judgment calls on, e.g., data cuts. Code should be legible and commented (give variables descriptive names; avoid long lines, etc.) Use the PEP 8 style guide as a guideline for formatting code. Avoid hard-coded variables, “magic numbers”, etc.

Use a separate notebook as “scratch space" for development and testing. However, **the notebook you turn in should be a polished product**. All cells should run sequentially; the final plots should be produced inline. Choose axis limits and scales carefully. Don’t leave old/broken code, pages of numerical output, or anything else that makes your notebook hard to read.

Write discussion and answers to questions in Markdown using the built-in LaTeX support for equations. Anything highlighted in pink must be turned in for credit. **You can consult with classmates, but all code, write-ups, and analysis must be your own. The use of AI is not permitted for submitted materials.** Refer to the syllabus for issues of collaboration, permissible use, and class conduct. When in doubt, ask.

For writing and testing your lab notebooks, you have several options. Creating notebooks locally on your computer is convenient at the cost of maintaining a working Python 3 installation[2]. Alternatively, you may use your CalNet ID to access Berkeley’s DataHub or Google Colab to host and access notebooks through your internet connection. We have also secured computing time on Savio—a high-performance computing cluster—for your use, particularly on Lab 3.

Most packages needed for the labs are already in the Astro Datahub environment and can be imported. You may create this same environment locally from `environment.yml` or `env_stable.yml` in our course materials repository. Additional packages may be installed from within the notebook environment using `pip`[3] If you need access to a computer in the undergraduate data lab, we can create you an account.

2. Technical components

- Databases & SQL
- Data visualization

[1] The *Jupyter Project* has deep roots here at Berkeley!

[2] Python 2 is fully depreciated.

[3] Lines of code beginning with ! execute as bash commands.

- Cluster-finding
- Stellar models
- Caching

3. Astronomy Background Material

You may refresh your knowledge on the relevant astrophysics with the textbook readings below:

- Carroll & Ostlie: parallax (p. 57), proper motion (p.16), magnitude system (p. 60), HR diagram (pp. 219, 475), stellar evolution (Ch 13), binary stars (Ch 7), extinction (p. 401)
- Ryden & Peterson: parallax (p.307), proper motion (p.445), magnitude system (p. 312), HR diagram (p. 346, 516), stellar evolution (Ch 17), binary stars (Ch 13.5), extinction (Ch 16)

4. Preamble

1. Make an account on the Gaia Archive by visiting https://gea.esac.esa.int/archive and clicking "SIGN IN" and "Register new user" in the upper right-hand corner. Having an account isn't necessary to query the Gaia catalogs, but it allows you to save previous queries and upload/query your own catalogs, which will be useful later.

   The Gaia Archive hosts most of the catalogs needed for this lab and Lab 1. To access a list of available catalogs, click "search", then "Advanced (ADQL)" tab on the upper left. Catalogs are listed on the left side of the page in nested drop-down menus.

   Use the Gaia Archive to test ADQL queries for errors before integrating them into a notebook.
2. Ensure `astroquery` is in your Python environment. The `astroquery.gaia` module allows you to combine Gaia ADQL queries with Python code.
3. Familiarize yourself with the Gaia mission. Below are papers and web resources, ordered roughly by relevance. You are not expected to read these in full, but some familiarity with the data products will prove helpful.
   (a) Babusiaux et al. 2018 (constructing color-magnitude diagrams with Gaia; arxiv: 1804.09378)
   (b) Brown et al. 2020 (general summary of the (early) 3rd Gaia data release; arxiv: 2012.01533).
   (c) Summary of the Gaia DR3 catalogs and the meanings of all columns.
   (d) Introduction to ADQL, the language (similar to SQL) for querying databases.
   (e) Lindegrin et al. 2020 (summary of Gaia astrometry; arxiv: 2012.03380).

5. Now the Fun: The Assignment

Construct color–absolute magnitude diagrams for the following star clusters: The Hyades (a young, nearby open cluster), M67 (an old, more distant open cluster), and NGC 6397 (a globular cluster, at an even larger distance). Babusiaux et al., 2018, provides an idea of what this means and how to accomplish it. The data you need can be found in the `gaiadr3.gaia_source` catalog.

Identify stars that are bonafide members of each cluster, distinguishing them from nearby stars that not physically associated. Check Gaia's astrometry (parallaxes and proper motions).

Section 3 of Babusiaux et al. provides guidelines for distinguishing cluster members from background stars. You needn't to follow their procedure exactly, and you may use other properties of these clusters from the literature, but you probably need cuts in position, parallax, and proper motion. Do *not* just search for the names of these clusters in the *Gaia* archive; that will return only a small fraction of the cluster members.

Perhaps try a cluster-finding finding algorithm to identify likely members based on proximity in phase space. This is likely the most principled option, but you can also do something more kludgy; e.g. circular or rectangular selections in position and proper motion.

Remove objects with unreliable photometry and astrometry. Read sections 2 and 3 of Babusiaux et al. 2018 for examples of quality cuts that filter out bad sources. Use less stringent cuts for objects with lower signal-to-noise ratios. Discuss what problems lead to bad astrometry and how the cuts you implement remove suspect objects.

With your final color–magnitude diagrams for cluster members, show (at least for one cluster) diagnostic plots that illustrate the selection of cluster members in phase space. For example, plot the proper motion in RA and Dec of potential cluster members based on position and parallax. Is there a clear clump of stars in proper motion space? Examine the color-magnitude diagram for potential cluster members with and without proper motion cuts. Do the cuts make the diagram cleaner? Similarly, show how astrometric quality cuts clean up the diagram.

Consider engineering a system for caching results, particularly for web queries. This safeguards against server outages[4] and avoids re-running time-intensive steps. Once you've run an ADQL query, you should be able to re-run the cell to retrieve the output without a web connection. Include a system for overwriting the cache when you change queries. Implementation is up to you, but one straightforward option is to save tables locally after running a query for the first time, with the option to re-run the query and overwrite the local copy when a boolean variable (e.g. `overwrite_cache=True`) is set. You can also use joblib, which will do most of the work for you.

Required checkpoint 1, due Wed Sep. 3, 2025: present (a) a color–absolute magnitude diagram for the Hyades, and (b) a plot showing the selection of members in proper-motion space. Submit this via gradescope.
It should be a pdf, named Firstname_Lastname_lab0_cp1.pdf

Once you have clean color–magnitude diagrams for the three clusters, overplot some synthetic photometry in Gaia bands from theoretical isochrones. Start with MIST (http://waps.cfa.harvard.edu/MIST/interp_isos.html) models. Look at models with a range of ages and metallicities. Determine which combinations of ages and metallicities are consistent with the data, and in doing so, estimate the age and metallicity of each cluster. Identify the various phases of stellar evolution in each cluster. Are there are stars that are likely to be in binary (or higher order) systems? Finally, comment on any discrepancies between the theoretical models and the data. While not required for the assignment, try comparing the MIST isochrones and best fits to the predictions from the PARSEC (http://stev.oapd.inaf.it/cgi-bin/cmd) models which are used in Babusiaux et al. 2018.

Required checkpoint 2, due Wed Sep. 10, 2025: present a color—absolute magnitude diagram for the Hyades that has MIST isochrones with a range of ages and metallicities overplotted. Submit via gradescope.
It should be a pdf, named Firstname_Lastname_lab0_cp2.pdf

6. Next Steps

Your full lab write up is due on 11:59 PM Sun Sep. 14, 2025. It should be a polished, executable Jupyter notebook with discussion of all plots and sections highlighted in magenta, named Firstname_Lastname_lab0.pdf. Run all cells (in order!) before saving as a pdf and submitting to Gradescope. Budget time wisely.

[4] Astronomy databases perversely go down for maintenance right before labs are due