Lab #0: Introduction to ADQL and Gaia Data

Astronomy Data Lab AY128/256 (UC Berkeley)

Assigned: Friday, August 23, 2024
Checkpoints: Monday, September 2, 2024; Friday, September 6, 2024
Final Write Up Due: Tuesday, September 10, 2024

1 Introduction & Logistics for Your Lab Submission

First of all: welcome to the **Astronomy Data Lab**! This first lab (indexed to zero of course!) serves as a gentle introduction to *Gaia* data and ADQL, a scripting language used to query databases. It is shorter than the three primary labs you'll be working on in this course, but it should provide a rough idea of the sort of work we expect you to do in Astro 128/256. We hope it will be useful to you in preparing for Lab #1. For this lab, the expectations for undergrads and grad students are the same.

For the 2 checkpoints, you will turn in a PDF rendering of a Jupyter notebook[1] that includes both discussion and the code and ADQL queries needed to reproduce your results. You should explain (in words) what your code and queries are doing and how you made any non-obvious judgement calls. Your code should be legible and adequately commented (e.g., give your variables descriptive names; avoid very long lines, etc.) You can use the PEP 8 style guide as a general guideline to help you make consistent looking Python code. Avoid hard-coded variables, "magic numbers", etc.

You may find it useful to use a separate notebook as a "scratch space" for testing and developing your code. In any case, **the notebook you turn in should be a polished product**. All the cells should already have been run sequentially, and the final plots should be produced inline. Your axis limits and scales should be chosen carefully. There should not be old code that didn't work mixed in with your final solutions, pages and pages of computer-generated numerical output, or anything else that will make your notebook hard to read.

You can write your discussion and answers to any questions in Markdown, using the built-in LaTeX support for equations. We have highlighted things in this lab that must be turned in for credit in pink. **You are welcome to consult with other students in the class, but all code, write-ups, and analysis must be your own. Also note the use of AI is not allowed for submitted materials.** Be sure to refer to the syllabus for issues of collaboration, permissible use, and class conduct.

You have several options for where to write and test your lab notebooks. If you have a working Python 3 installation[2] and know how to create notebooks locally, that should work fine. Alternatively, with a CalNet ID you can use Berkeley's Datahub (http://astro.datahub.berkeley.edu) or Google Colaboratory (http://colab.research.google.com) to host and run notebooks that can be accessed from anywhere with an internet connection (even without a local Python installation).

Most of the packages needed for the labs are already available at our Astro Datahub and can be imported immediately. If you want to install new packages, this is also straightforward (you can execute bash commands from within the notebook environment). If you would like access to a computer in the undergraduate data lab, please let us know and we will have an account created for you.

[1] BTW, the *Jupyter Project* has deep roots here at Berkeley...
[2] We will insist that you use Python 3 for this class. Python 2 support in Jupyter, numpy, scipy, etc is depreciated!

2 Technical components

- Databases & SQL
- Data visualization
- Cluster-finding
- Stellar models
- Caching

3 Astronomy Background Material

Below is an extensive (but not complete) list of textbook readings you can do to refresh your knowledge on some of the basic astrophysics required for this lab.

- 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 yourself an account on the Gaia Archive by going to https://gea.esac.esa.int/archive and clicking "SIGN IN" and "Register new user" in the upper right-hand corner. You don't need an account to query the Gaia catalogs, but having an account will allow you to save your previous queries and to upload and query your own catalogs, which will be useful later on.

   The Gaia Archive hosts most of the catalogs we will need for this lab and Lab #1. To see which catalogs are available, click "search", and click on the "Advanced (ADQL)" tab on the upper left. Available catalog are listed on the left side of the page in nested drop-down menus.

   The Gaia Archive is a good place to test out your ADQL queries before integrating them into your notebook. If your queries have errors, the error messages in the Gaia Archive will help you identify them.

2. Install `astroquery` in your Python environment[3]. The `astroquery.gaia` module will allow you to combine Gaia ADQL queries with Python code.

3. Familiarize yourself with the Gaia mission. Here are some papers and web resources for doing so, ordered roughly by their usefulness to you for this assignment and Lab #1. You are not required or expected to read these in full, but some familiarity with the data products will be extremely very helpful.

   (a) Babusiaux et al. 2018 (about construction of 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).

[3] For those using Datahub, astroquery, should have already been installed.

(c) Summary of the Gaia DR3 catalogs and the meanings of all columns: https://gea.esac.esa.int/archive/documentation/GDR3/Gaia_archive/chap_datamodel/sec_dm_main_source_catalogue/ssec_dm_gaia_source.html

(d) Introduction to ADQL, the database language used to query the catalogs (this is quite similar to SQL): https://www.cosmos.esa.int/web/gaia-users/archive

(e) Lindegrin et al. 2020 (summary of Gaia astrometry; arxiv: 2012.03380).

5 Now the Fun: The Assignment

After much ado, here's the meat of 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). If you've looked at Babusiaux et al. 2018, you should have an idea of what this means and how to accomplish it. Some hints are below. The data you need can be found in the `gaiadr3.gaia_source` catalog.

You will need to identify stars that are actually members of each cluster, and to distinguish them from other stars that are in the same part of the sky but not physically associated with each cluster. Gaia astrometry (parallaxes and proper motions) will be useful for this.

Section 3 of Babusiaux et al. provides some guidelines for distinguishing cluster members from background stars. You don't need to follow their procedure exactly, and you can make use of any known properties of these clusters in the literature, but your selection should involve some sort of cuts in position, parallax, and proper motion. You should *not* just search for the names of these clusters in the *Gaia* archive; that will return only a small fraction of cluster members.

One option is to use a cluster-finding finding algorithm to identify likely cluster members (e.g. based on the number of nearby neighbors they have in phase space). If you know how to implement such a search, it is likely the most principled option. For this assignment, it's also fine to do something more kludgy; e.g. circular or rectangular selections in position and proper motion.

You will want to remove objects with unreliable photometry and astrometry. Read through sections 2 and 3 of Babusiaux et al. 2018, and use quality cuts similar to those they recommend to filter out bad sources. You will probably want to use less stringent cuts for objects that are relatively far away (and thus have lower signal-to-noise ratios) than for those that are nearby. Discuss what sorts of problems lead to bad astrometry and how the cuts you implement remove suspect objects.

Along with the final color–magnitude diagrams for cluster members, show (at least for one cluster) some diagnostic plots that illustrate the selection of cluster members in phase space. For example, try plotting the proper motion in RA and Dec of potential cluster members based on a selection in position and parallax only. Is there a clear clump of stars in proper motion space that corresponds to the cluster? Try plotting the color-magnitude diagram for potential cluster members with and without cuts on proper motion. Do the proper motion cuts make the color-magnitude diagram cleaner? Also, show how your astrometric quality cuts clean up the final color-magnitude diagram (i.e., compare the CMDs you obtain with and without quality cuts).

Here and in all future labs, you should develop a system for caching[4] **the results of web queries locally.** This is a safeguard in case, say, the *Gaia* archive crashes an hour before the lab is due[5]. **That is, once you've run an ADQL query once, you should be able to run the cell that retrieves its output again, without a web connection.** You'll also want to implement a system for

[4]This means saving important files locally so you don't have to keep re-running your queries, which can get time intensive.
[5]Astronomy databases tend to go down for maintenance right before our labs are due

overwriting the cache when you change your queries. How you implement the cache is up to you. One straightforward option is to save the tables returned from queries locally after you run a query for the first time, and to only re-run the query and overwrite the local copy when you manually set a boolean variable (e.g. `overwrite_cache=True`). You can also use joblib, which will do most of the work for you.

Required checkpoint 1, due Monday 9/02/24 by 11:59pm: 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.

Required checkpoint 2, due Friday 9/6/24 by 11:59pm: present a color—absolute magnitude diagram for the Hyades that has MIST isochrones with a range of ages and metallicities overplotted. Submit this via gradescope.
It should be a pdf, named Firstname_Lastname_lab0_cp2.pdf

Once you have done this, try comparing to the predictions of PARSEC (http://stev.oapd.inaf.it/cgi-bin/cmd) models instead of MIST. Comment on any significant differences.

6 Next Steps

Your full lab write up is due on Tuesday 9/10/24 by 11:59pm. It should be uploaded as a written report (in LaTeX) in PDF format (not a PDF version of a Jupyter notebook), named Firstname_Lastname_lab0.pdf. Be aware that writing up takes a lot longer than you think. Budget your time wisely. A significant portion of class time along the way is set aside for interactive discussion of your progress. Come to class on Monday Sept 9 prepared.