Eppur si muove

Edgar Ortiz
edgar.ortiz@ua.cl

Abstract

In this work, I have developed python code to show how does the sky looks using data from the Gaia DR2. The illustrations are presented in an aitoff projection in galactic coordinates. Since the Gaia DR2 has over a billion sources, I limit my data to the 250.000 brightest sources. Throughout this work the **astropy.coordinates** module, as well as **matplotlib**, are used to produce:

1. An image of the sky with these 250.000 stars,
2. A map of the 5000 brightest stars, where the color of each star indicates its radial velocity.
3. An image of how the sky will look from three different stars: Sirius, Canopus and Rigil Kentaurus. These three images have a color bar attached to indicate the magnitude of the stars.
4. Finally, I created 10 seconds moving showing the evolution of the sky for the 5000 brightest sources, over a period of 10.000 years. The movie has 60 frames per second ratio.

Introduction

Gaia space telescope’s main goal is to build the largest and most accurate catalog of stars' positions and velocities (DR2 is already available) in the Milky Way, see: https://sci.esa.int/web/gaia. In this work, I want to use part of Gaia DR2 in conjunction with python, to generate visual representations, to show to the general public the dynamics of the stars observed by Gaia. To achieve this goal, I make use of the python package 'astropy.coordinates', which provides functionality for handling celestial/spatial coordinates and their velocities with ease. This package comes in handy to manipulate the data we have from Gaia in order to obtain the visualizations I stated in the abstract.

Data

To obtain the set of visuals I want to show to the public, I made use of the ADQL language provide in the Gaia website to query for the 250.000 brightest sources in the archive. The instructions used were:

```
select top 250000 ra, dec, pmra, pmdec, parallax, radial_velocity,
phot_g_mean_mag
from gaiadr2.gaia_source
where abs(pmra)>0
or abs(pmdec)>0
order by phot_g_mean_mag
```

This generates a VO-table with the following data:

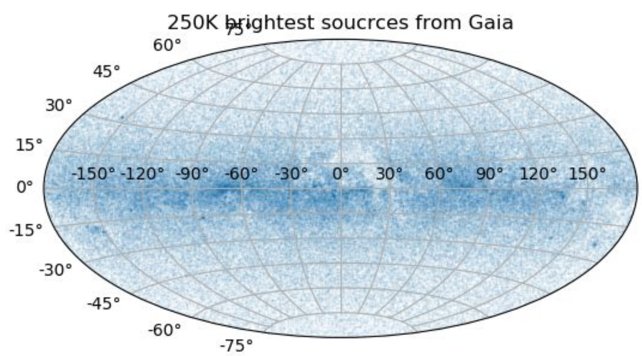
- **ra, dec**: right ascension and declination in [deg].
- **pmra, pmdec**: proper motion in right ascension and declination direction in [mas/year] (angular velocity)
- **parallax**: the absolute stellar parallax in [mas]
- **radial_velocity**: spectroscopic radial velocity in the solar barycentric reference frame in [km/s]
- **phot_g_mean_mag**: G-band mean magnitude in [mag]

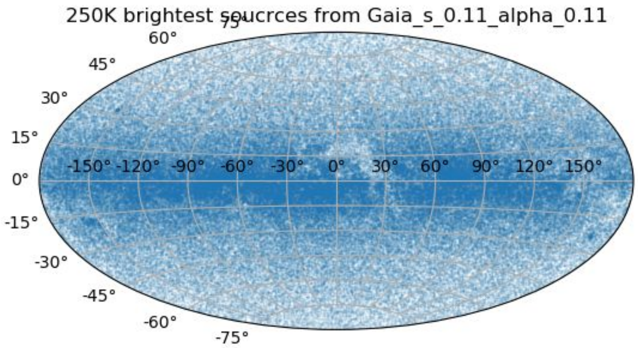
Science questions, code & results

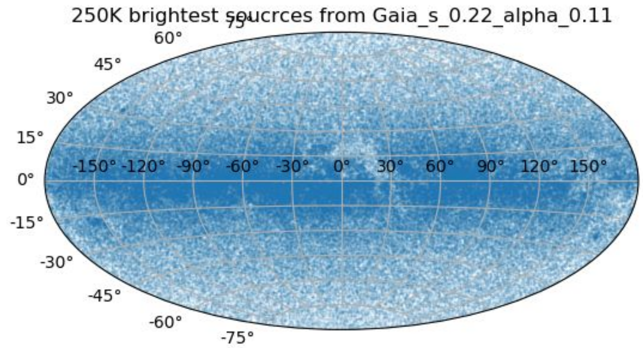
As mentioned before, all the illustrations that I generated in this work, correspond to maps in an aitoff projection, using galactic coordinates. For some illustrations, a color bar is needed to illustrate either the radial velocity or the magnitude for a given set of stars. To accomplish that, I had to create a customizable color bar that could adapt to the data at hand.

1. **Image of the Sky:** The plots below show the distribution of the 250 thousand stars over the sky, they are plotted in blue. As we can notice, most of the stars in this set lye at zero degrees in the horizontal. The function used to generate this plot has two parameters, s, and alpha, controlling the shape of the source marker and the blending value. Since we have many sources, the marker size, and the blending value have to be small.

s=0.05, alpha = 0.05

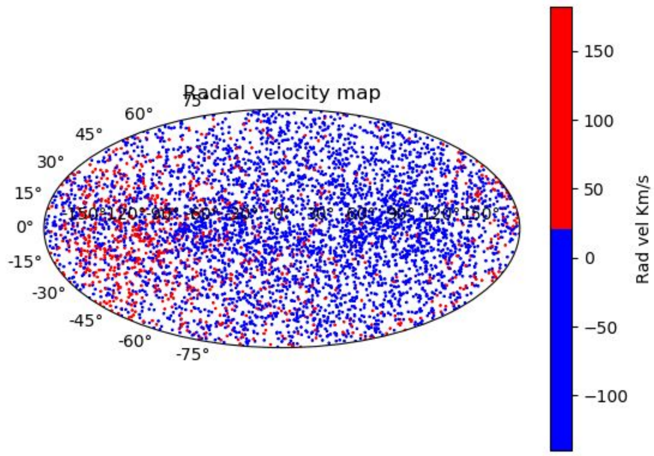


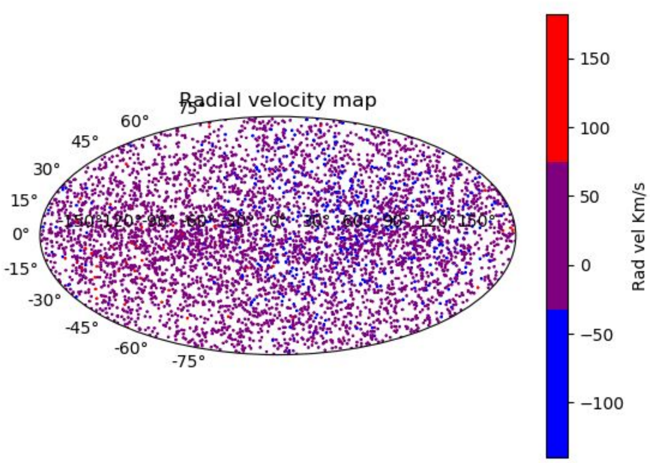


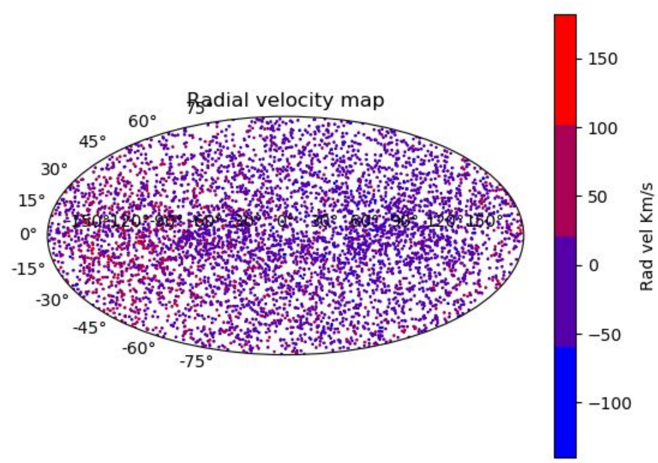


2. **Radial velocity map:** As for this case, I generate an aitoff plot, too. The main differences here are that it is only for the 1000 brightest stars in my data. During the process, I found that there were sources with missing values for the radial velocity. To account for that, I had to take them out and select only the 1000 brightest sources with radial velocity available. I implemented a color bar that runs from blue to red, meaning stars in blue moving towards us and stars in blue are receding. Since there are a plethora of velocities, I generated different plots to illustrate some variations.

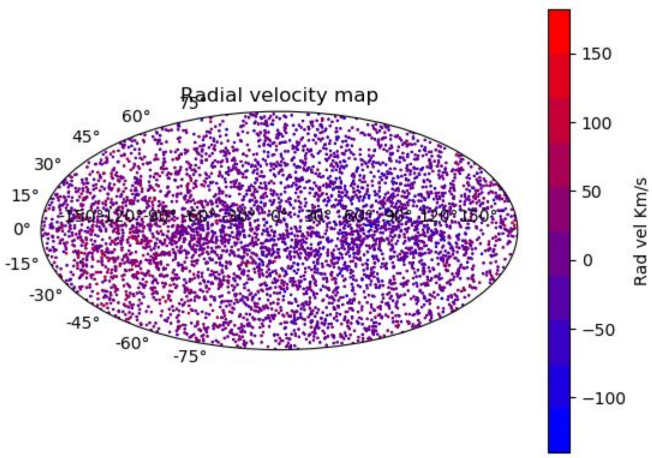
   In this first three plots, having a less sensitive color bar, we can inspect and gain a general sense of whether a star moves towards us or if it is receding

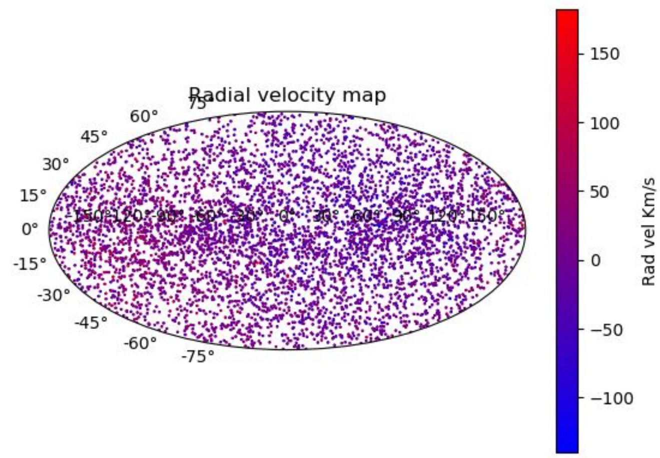

Radial velocity map
75°
60°
45°
30°
15°
0°
-15°
-30°
-45°
-60°
-75°
150°
100
50
0
−50
−100
Rad vel Km/s


Radial velocity map
75°
60°
45°
30°
15°
0°
-15°
-30°
-45°
-60°
-75°
150°
100
50
0
−50
−100
Rad vel Km/s


Radial velocity map
75°
60°
45°
30°
15°
0°
-15°
-30°
-45°
-60°
-75°
150°
100
50
0
−50
−100
Rad vel Km/s

And finally, an even more refined color bars giving us more precise information about the distribution of the radial velocities:

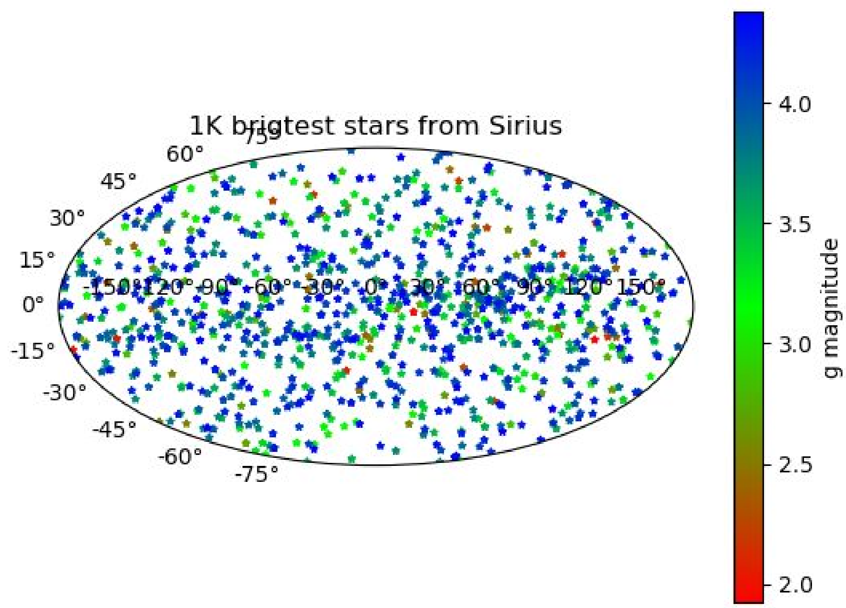




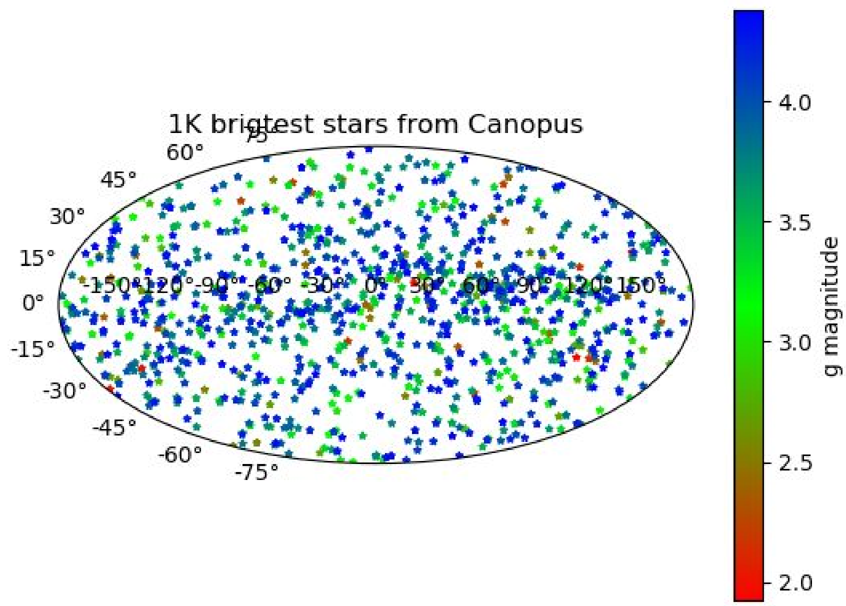
3. **Other skies:** in this case, I generated an image of how the sky would look like for someone living in a planet that is orbiting the next three stars: Sirius, Canopus and Rigil Kentaurus. I chose these three stars because they are the brightest stars you can see from Earth, after the sun, obviously. In this case, I considered only the 1000 brightest stars and I assumed that the magnitude of these stars remains unchanged when we change to the new origin.
   To accomplish this task, first I generated a coordinate frame in galactic coordinates, having as its center each of the mentioned stars. After that, I used the method **transform_to** from the object **SkyCoord** to pass the coordinates of the 1000 selected stars to the new reference frame. As an example, here I provide the lines of code I used for Sirius:

```
# setting the center
sirius_lon = Angle(227.23028548*u.degree).wrap_at(180*u.degree)
sirius_lon = sirius_lon.value *u.degree
sirius_lat = -08.89028243 *u.degree
center = SkyCoord(l = sirius_lon, b =sirius_lat, frame = 'galactic')
sirius_frame = center.skyoffset_frame()
# transforming the positions to the new reference frame
target = SkyCoord(l, b, frame='galactic')
target = target.transform_to(sirius_frame)
```

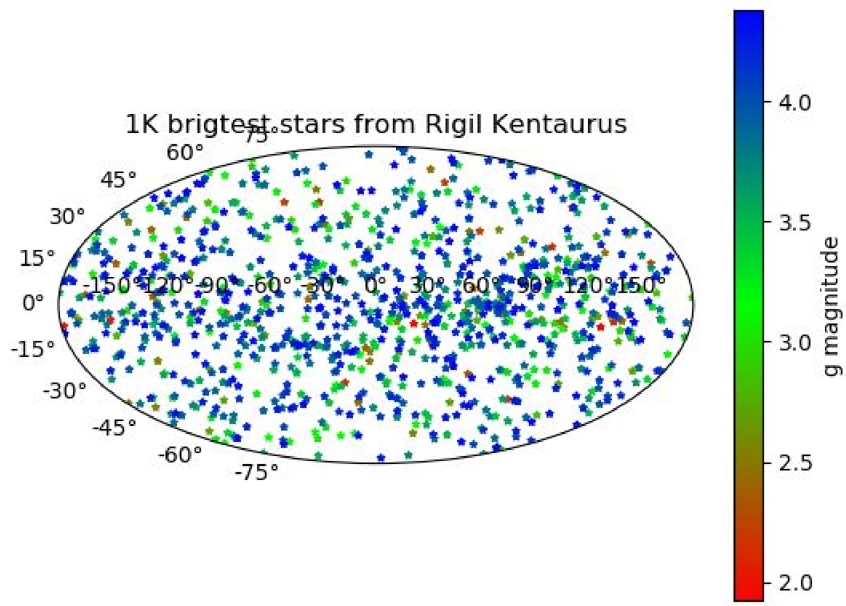
I also used a color bar for the magnitude of the 1000 stars to have a broader context on how different the sky looks like. Here the results for Sirius:



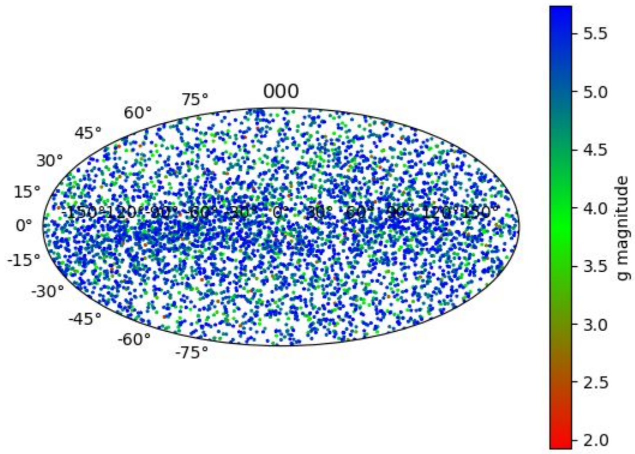
Here the results for Canopus:

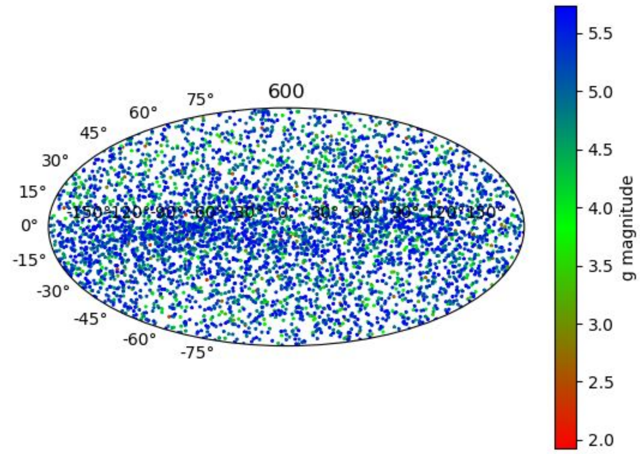


And finally, here the results for Rigil Kentaurus



4. **Sky evolution:** In this case, I took the 5.000 brightest stars to generate a small video showing the movement of the stars in a time period of 10.000 years. The video is 10 seconds long and it has 60 frames per second ratio. When running the code to generate the motions, I got an error because the data had some negative values for the parallax. Therefore I selected the same amount of stars but all with positive parallax. In this case, it was extremely useful the method **apply_space_motion** of the class **SkyCoord.** Here as before, I also added a color bar for the magnitude so we can better visualize the motion of the stars. The images below show the sky before applying any motion and after 10.000 years, from left to right (labeled 000 and 600):





Conclusions

The final output of this work consists of a set of visualizations that can be shown to made the imagination of the general public go nuts about the dynamics of our galaxy. In a more technical fashion, I'd like to comment on how useful is OOP and how astropy has used it to make packages like this, **astropy.coordinates;** that saves huge amounts of time when it comes to playing around with data.