Astroinformatics
Proceedings IAU Symposium No. 325, 2016
M. Brescia, S.G. Djorgovski, E. Feigelson,
G. Longo & S. Cavuoti, eds.

doi:10.1017/S1743921316012795

Interactive (statistical) visualisation and exploration of a billion objects with vaex

Maarten A. Breddels

Kapteyn Astronomical Institute, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands
email: `breddels@astro.rug.nl`

Abstract. With new catalogues arriving such as the *Gaia* DR1, containing more than a billion objects, new methods of handling and visualizing these data volumes are needed. We show that by calculating statistics on a regular (N-dimensional) grid, visualizations of a billion objects can be done within a second on a modern desktop computer. This is achieved using memory mapping of hdf5 files together with a simple binning algorithm, which are part of a `Python` library called `vaex`. This enables efficient exploration or large datasets interactively, making science exploration of large catalogues feasible. `Vaex` is a `Python` library and an application, which allows for interactive exploration and visualization. The motivation for developing `vaex` is the catalogue of the *Gaia* satellite, however, `vaex` can also be used on SPH or N-body simulations, any other (future) catalogues such as SDSS, Pan-STARRS, LSST, etc. or other tabular data. The homepage for `vaex` is `http://vaex.astro.rug.nl`.

1. Introduction

Gaia is an European Space Agency (ESA) cornerstone satellite mission, that aims to measure accurate astrometry (sky positions, parallax and proper motions) for over a billion stars in the Milky Way. Compared to the previous Hipparcos satellite (Perryman *et al.* 1989), which measured $\sim$ 120 000 parallaxes accurately, the *Gaia* data is expected to revolutionize our knowlege of the Milky Way. The *Gaia* satellite was launched on 19 December 2013 (Gaia Collaboration, *et al.* 2016), and we recently had its first data release of over a billion sources (Gaia Collaboration *et al.* 2016; Lindegren *et al.* 2016). Although not all sources have their five astrometric properties determined, the positions and G band fluxes/magnitudes (van Leeuwen *et al.* 2016) are available for every object in the catalogue.

Working with a catalogue of a billion objects is not an easy task. However, for many science cases, as well as quality checks of the data, we would like to visualize all or large parts of the data. While scatter plots would suffice when working with the Hipparcos catalogue, doing the same for the full *Gaia* catalogue, would not be useful. Apart from the long time it takes to render each individual point as a glyph, the overplotting makes the plot meaningless, as we demonstrate in Fig. 1. In this figure, we show how plotting a random subset of 10^4 stars (top left panel) shows structure in the galactic disk, while plotting 10^6 stars (top right panel) already starts to hide any structure that is present in the data, due to overplotting. In the bottom panel of this figure, we show the density of points on a grid, with low densities corresponding to black, and high densities to white. This visualization shows much more structure in the data, such as the dust that is present in the disk, as well as artifacts in the data due to the scanning nature of the satellite. Preferably, we would like to have an interactive version of this visualization, where one would be able to zoom and pan, but also to select a region (for instance the

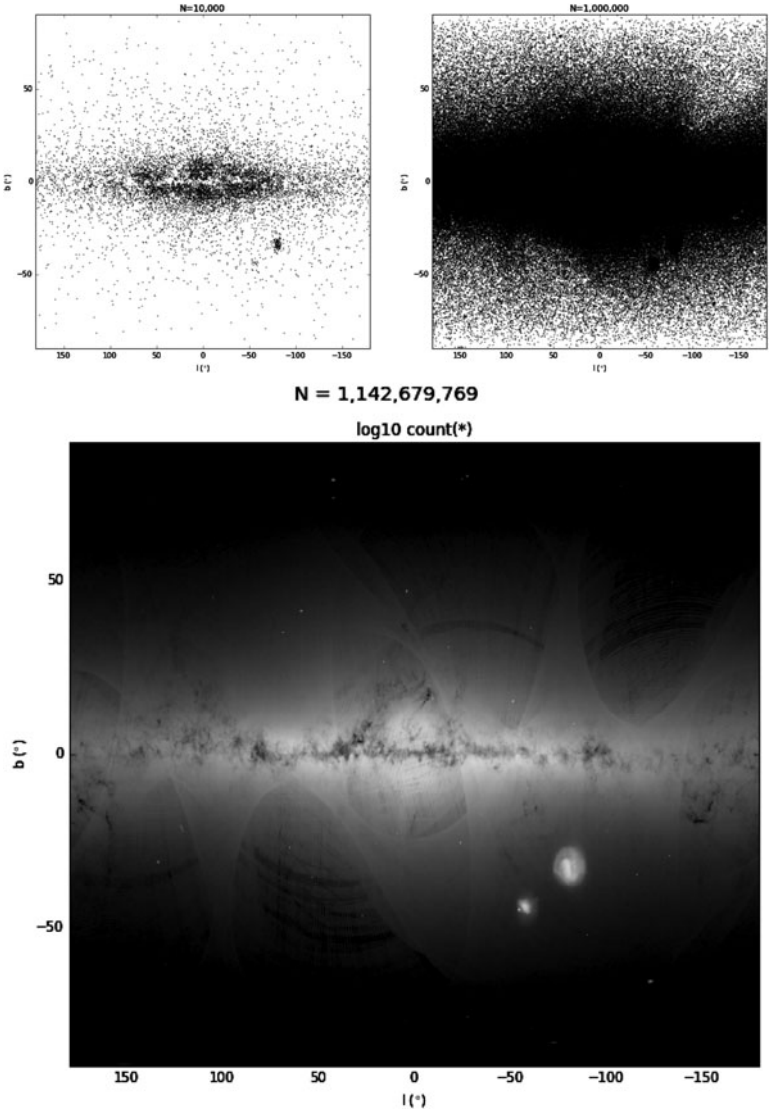

Figure 1. Illustration of scatter plot versus a density plot in galactic coordinates of the *Gaia* DR1 catalogue showing how a scatter plot can fail, while a density plot shows the rich structure in the data. **Top left:** Scatter plot showing 10 000 points. **Top right:** Idem, with 1 000 000 points. **Bottom:** Density plot with 1 142 679 769 points. The top left and bottom plot show more structure in the galactic disk compred to the top right, where overplotting hides details, and the density plot shows even more structure, such as artifacts in the data due to the scanning nature of the satellite. The bottom visualization can be generated in less than a second.

Large Magellanic Clouds), and make other visualizations of this selections (for instance a histogram of G magnitudes).

No software package that we know of can currently handle this. While `TOPCAT` (Taylor 2005) can do many of these interactive visualizations and selections, it does not scale to a billion rows. The datashader library (`https://github.com/bokeh/datashader`) can do the interactive visualization of a billion rows (although slower compared to our software), but does not provide mechanisms for efficient selections and focusses on 2d.

https://doi.org/10.1017/S1743921316012795 Published online by Cambridge University Press

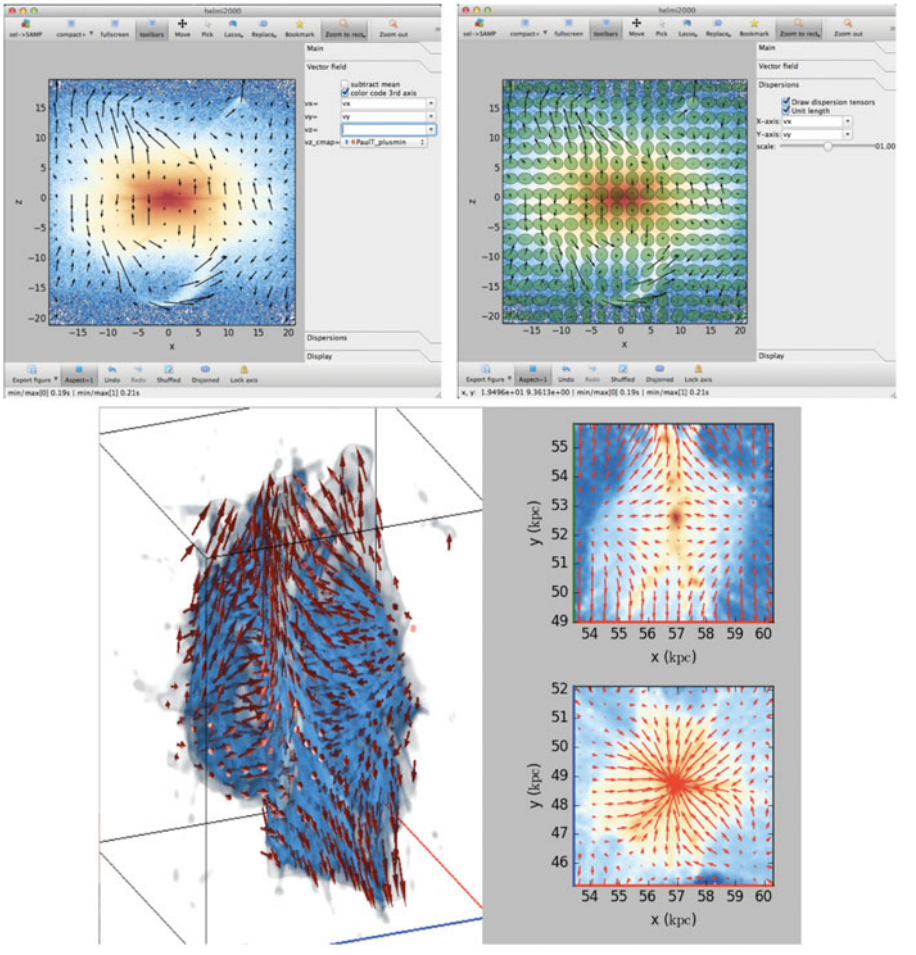

Figure 2. Illustration of visualization of statistics (mean velocity) using vectors (top left panel), a velocity dispersion tensor using ellipses (top right panel), or the mean velocity in 3d using vectors (bottom panel).

2. Main ideas

In order to do the visualization of a billion rows interactively, we would like to be able to generate a visualization as in the bottom panel of Fig. 1 in about 1 second. If we take this as an example, where we need to process $\sim 10^9$ rows and two columns (galactic longitude l and galactic latitude b) of double precision (8 bytes per double), we need to bring a total of $10^9 \times 2 \times 8$ bytes $= 16 \times 10^9$ bytes $= 16$ GB ≈ 15 GiB of data to the CPU. With current desktop machines having a memory bandwith of $\sim 10 - 30$ GB/s this poses no problem.

Futhermore, with a quadcore CPU of 3 GHz, this leaves 12 CPU cycles/row/second, which is only enough to do really simple operations. We therefore only consider doing simple statistics (counts, sums, maximum, minimum) on a regular grid. With these simple algorithms, many statistics can be calculated in the order of a second, which then can be visualized. Examples are: Calculating the counts on a regular grid (right panel of Fig. 1), the mean velocity represented by vectors (top left panel and bottom panel of Fig. 2),

https://doi.org/10.1017/S1743921316012795 Published online by Cambridge University Press

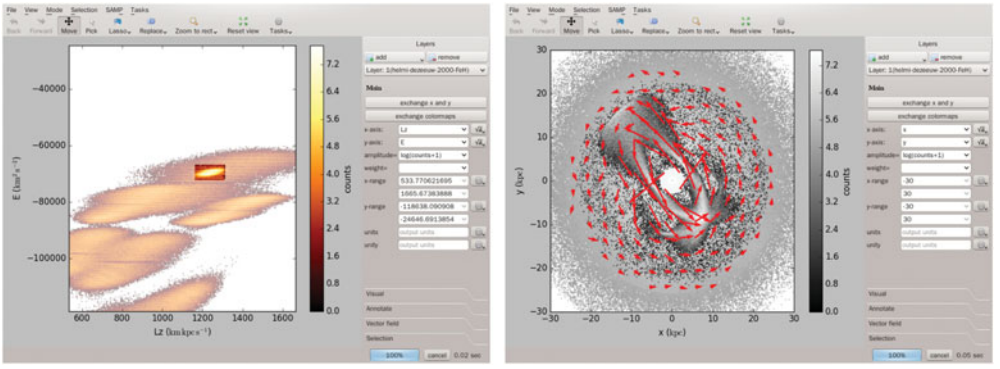

Figure 3. Left: 2d plot window, showing E vs L_z with a cluster in this space selected. **Right:** 2d plot windows, showing y vs x, sharing the same selection as the other window, i.e. linked views.

velocity dispersion tensor (top right panel of Fig. 2), total flux, correlation between velocity components, etc.

In order to get this performance, the data should be able to fit into main memory, otherwise the performance is limited by the storage device. To avoid unnecessary memory copies, we store the data in hdf5 files, in a column oriented way, and memory map this data.

3. Implementation

These ideas are implemented in a `Python` package called `vaex`, with the core statistical algorithms being implemented in the `C` language. This library takes care of reading of the data, multithreading, the statistical calculations and efficient implementations of doing selections, and visualization based on matplotlib (Hunter 2007), or OpenGL (for the 3d rendering). The statistical algorithms work in N dimensions†, on either the full dataset, or a selection. Each dimension of the regular grid is defined by a column, or mathematical operations on it, the number of bins in each dimension, and the coordinates of the begin and end bin. `Vaex` is open source (MIT License), the main code repository can be found on Github ‡, and its homepage is `http://vaex.astro.rug.nl`. Part of the `vaex` software, is a Graphical User Interface (GUI) build with Qt, that enables interactive visualization (zooming and panning) and exploration (selections/queries). This last part is also distributed as a standalone software package, available for Linux and OSX. The Python package is available as source (from github), or as more easily installable pip or conda package (see the webpage for more installation details).

4. Examples

For demonstation purposes, we include a random 10% of the data set from Helmi & de Zeeuw (2000) with the application. This dataset contains a sample of 33 simulated satellites galaxies which are disrupted in a Galactic-like halo. While most of the satellites are fully phase mixed, and not distinguisable in configuration space, there are prominent clumps present in the space of energy (E) and angular momentum around the z-axis (L_z). In Fig. 3 we show the E-L_z space in the left panel, and made a selection in this space,

† Where zere dimensional would be a scalar value, such as the mean of a column.
‡ `https://github.com/maartenbreddels/vaex`

https://doi.org/10.1017/S1743921316012795 Published online by Cambridge University Press

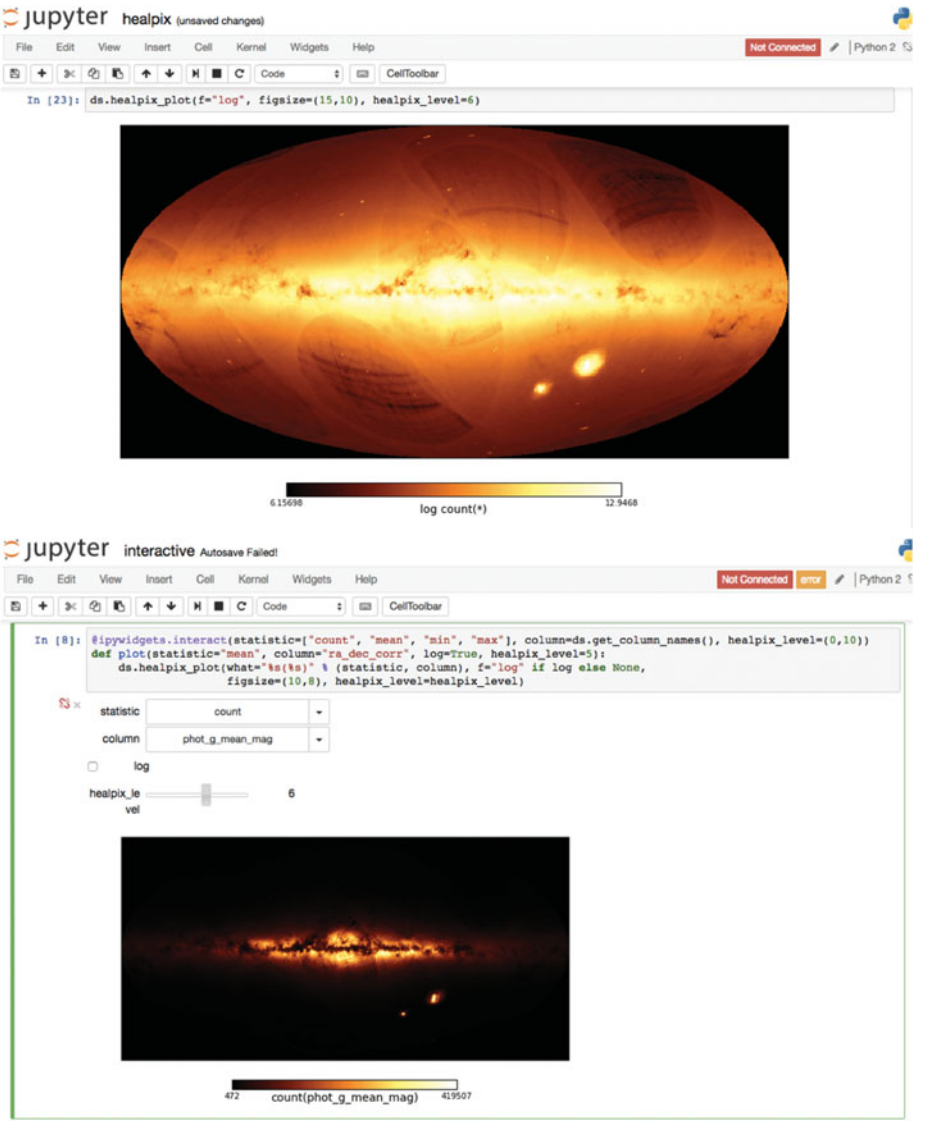

Figure 4. Top: Visualization using statistics on a healpix index to plot the full *Gaia* DR1 catalogue sky distribution. **Bottom:** Using ipywidgets, drop down menus and checkboxes can be linked to the visualization to enable custom plots with interactivity.

while on the right panel we show the corresponding selection in configuration space (x-y projection) and its mean velocity using vectors. Here we see this clump in E-L_z space corresponds to a stream that is not yet fully phase mixed, and the visualization of the velocities shows its coherent space motion. This linking between the selection in multiple visualization is commonly called 'linked views'.

To visualize the billion rows of the *Gaia* DR1 catalogue, we can run vaex in server mode on a machine which has enough memory to contain many columns in main memory, and connect to it using the vaex program. Now we can visualize the full *Gaia* DR1 from even low end machines or laptops. For instance, the visualization shown in the bottom panel of Fig. 1 showing the full *Gaia* DR1 catalogue can be generated in less than as second.

https://doi.org/10.1017/S1743921316012795 Published online by Cambridge University Press

Apart from the GUI, the `vaex` library is more powerful in the Jupyter notebook (Pérez & Granger 2007), where the full Python programming language can be used to customize computations and visualization. In the left panel of Fig. 4 we show that in the notebook it is possible to visualize an all sky plot using statistics on a healpix index. In the bottom panel of the figure, we show that by using the ipywidgets† library, with minimal effort, interactive options can be added to create custom plots.

5. Conclusions

The `vaex` library can handle 10^9 rows per second to calculate statistics on a (N-dimensional) regular grid. These statistics can be visualized for 1, 2 and 3 dimensions in the vaex program, and in the Jupyter notebook. The performance allows for interactive visualization (zoom and pan) and exploration (selections/queries) of massive catalogues such as *Gaia* DR1. This allows one to fully exploit not only *Gaia*, but also upcoming catalogues of similar or largers such as Pan-STARRS (Kaiser *et al.* 2010), LSST (Ivezic *et al.* 2008), etc. or any other tabular data such as SPH or N-body simulations. Not only will `vaex` enable interactive visualization and exploration of large catalogues, but the fast performance will also stimulate the exploration of ideas otherwise hampered by computational time or resources.

Acknowledgements

MB thanks Amina Helmi for making this work possible, not just financially. MB also thanks Jovan Veljanoski for his feedback on the Python API making it more human friendly. MB acknowledges financial support from NOVA. This work has made use of data from the European Space Agency (ESA) mission *Gaia* (`http://www.cosmos.esa.int/gaia`), processed by the *Gaia* Data Processing and Analysis Consortium (DPAC, `http://www.cosmos.esa.int/web/gaia/dpac/consortium`). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the *Gaia* Multilateral Agreement.

References

Gaia Collaboration, Brown, A. G. A., Vallenari, A., *et al.* 2016, ArXiv e-prints, arXiv:1609.04172

Gaia Collaboration, *et al.* 2016, ArXiv e-prints, arXiv:1609.04153

Helmi, A., & de Zeeuw, P. T. 2000, *MNRAS*, 319, 657

Hunter, J. D. 2007, *Computing In Science & Engineering*, 9, 90

Ivezic, Z., Tyson, J. A., Abel, B., *et al.* 2008, ArXiv e-prints, arXiv:0805.2366

Kaiser, N., Burgett, W., Chambers, K., *et al.* 2010, in Proc. SPIE, Vol. 7733, Ground-based and Airborne Telescopes III, 77330E

Lindegren, L., Lammers, U., Bastian, U., *et al.* 2016, ArXiv e-prints, arXiv:1609.04303

Pérez, F., & Granger, B. E. 2007, *Computing in Science and Engineering*, 9, 21

Perryman, M. A. C., Hassan, H., Batut, T., *et al.*, eds. 1989, The Hipparcos mission. Pre-launch status. Volume I: The Hipparcos satellite., Vol. 1

Taylor, M. B. 2005, in Astronomical Society of the Pacific Conference Series, Vol. 347, Astronomical Data Analysis Software and Systems XIV, ed. P. Shopbell, M. Britton, & R. Ebert, 29

van Leeuwen, F., Evans, D. W., De Angeli, F., *et al.* 2016, A&A special Gaia volume

† `https://github.com/ipython/ipywidgets`

https://doi.org/10.1017/S1743921316012795 Published online by Cambridge University Press