

GlueViz Exercises

J H Pickering

J.H.Pickering at leeds.ac.uk

University of Leeds

1. Running GlueViz

Opening GlueViz produces the window show in figure 1. There are four separate areas in the user interface.

- A. **Data Collection:** in this area the data sources you have opened, and the subsets you have defined on the data are listed.
- B. **Plot Layers:** in this area the data that are available in the plot you are currently working on are listed, you may have options about how and which data are displayed.
- C. **Plot Options:** in this area the options for the currently active plots are available.
- D. **Plotting:** in this, the largest, area the various visualizations are displayed.

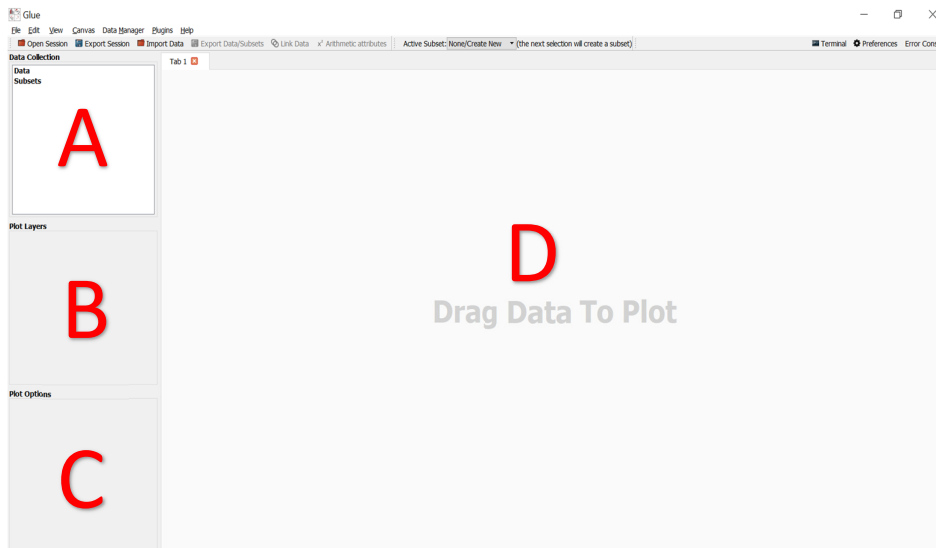


Figure 1 The layout of the GlueViz main window. At the top there is a tool bar below which there are four areas: Data Collection area A, all the open data sources and subsets; Plot Layers area B, the data used in the currently active plots; Plot Options area C, the display options for the currently active plots; Plotting area D, where all the plots appear.

The three groups of buttons at the top are the main way of interacting with data. Their functions are described in figure 2, 3 and 4.

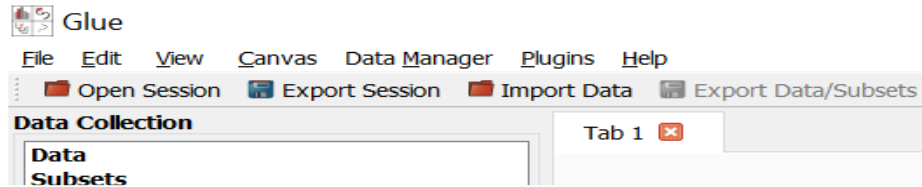


Figure 2 File input-output buttons: 'Open Session' load an old GlueViz session; 'Export Session' save your current work; 'Import Data' load a data file; 'Export Data/Subsets' save data or subset in comma separated file.

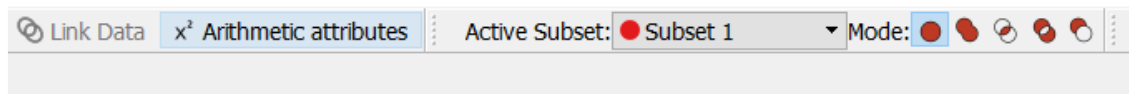


Figure 3 Data manipulation controls: 'Link Data' connect data in different files/tables; 'Arithmetic attributes' compute new attributes based on existing; 'Active Subset' select the active subset and decide how it is to be changed by further selections (replace, union, intersection, anti-intersection, subtraction).

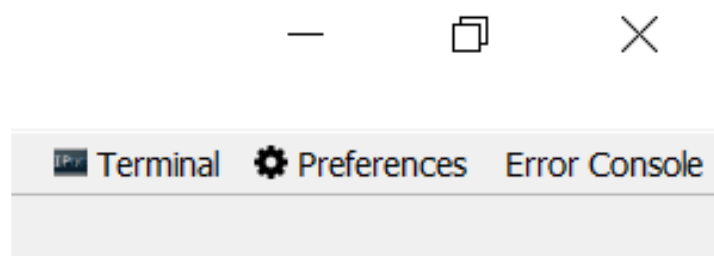


Figure 4 Other controls: 'Terminal' open an IPython terminal for scripition; 'Preferences' general appearance of plots; 'Error Console' list of Python errors generated.

The following working is adapted from the tutorials on the GlueViz site, from which the data used in the course was sourced.

2. Loading, Displaying and Selecting

2.1. Loading and Displaying

On the file Menu select "Open Data", then navigate to the sample data and select "w5.fits".

When the data set has appeared in the data collection area A, click on it, and drag it to the plotting area D, drop and select '2D Image Viewer' from the pop-up.

To make the image to fill the plotting area, click the maximize button, the resize controls are now very small, at the top right-hand corner above the label "Error Console", see figure 5.

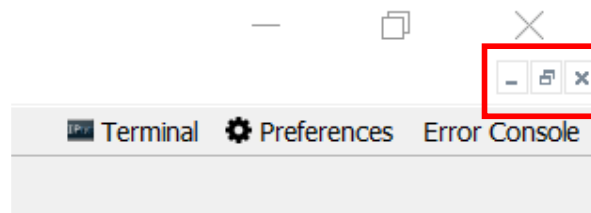


Figure 5 The location of the plot Minimize, Restore and Exit controls, when the plot is maximized.

On the left the Plot Layers area now holds the settings used to generate the image from the raw astronomical data.

1. Try changing the setting Linear to Logarithmic, this changes the way pixel intensities in the image are calculated from values in the data.
2. Try changing the limits on the data used to generate the image to 95%.
3. You can also play with the contrast and bias of the image.

On the left the Plot Options area allow you change overall features of the plot for example on the Axis tab you can change the size of the labels on the axis

There is a series of buttons on the bar immediately above the image, these allow you to manipulate or save the image.

Try using the Zoom to rectangle button (magnifying glass icon) to zoom a region. You can reset using the home button (house icon).

Most important they allow you to select subsets of the data.

WARNING: In the versions I have in conda the ability to move a subset by holding the control key, then clicking and dragging is not working.

1. Use the rectangle button to select a small region of the image. The new subset will appear in the data collection area the new subset will appear.
2. In plot layers there is a second entry for the subset. Try checking and unchecking the sets and see the results. Also note you can still move the subset at this stage.
3. You can now add extra constraints to the subset.
 - a. To do this ensure that the active subset selector to centre top is set to the intersect.
 - b. Then use the mode to set the way the new constraint will interact with the old; 'add', 'intersection' and 'remove' are the most useful, see figure 6.
 - c. You can now draw a second selection which will interact with the first, we are still producing only one subset.

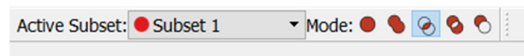


Figure 6 Subset 1 selected, with the option for the next operation to modify subset 1 by intersecting the new user input with subset1.

2.2. Saving Your Current Work

To save your current work click the “Export Session” button, the one from the File menu has a bug for some data so try to avoid using it.

You can reload it later using the ‘Open Session’ button, or when starting from the command line by typing “glue <file name>.glu”. There are three options available when saving:

1. **Relative paths:** the source files addresses will be stored relative to the directory from which you are running GlueVis.
2. **Absolute paths:** the source files addresses are recorded relative to the computer so you can start GlueVis from a different file. May be useful if you are programming and wish to run in different locations.
3. **Including data:** the raw data is including in the export. This is useful if you need to send your work to someone else, but it can generate very large files.

WARNING: In the versions I have in conda closing a plot window often causes a crash so frequent saving is a good idea.

2.3. Alternative Views

You can try other views using the subset you have created.

1. From the Plot Layers drag the subset to the plotting area D, then select a histogram to display the data.

2. Note in Plot Layers area you now only see the subset, and in the Plot Options area you only see the factors relating to the histogram.
3. Try selecting the image and the histogram in turn and you will see the Plot Layers and Plot Options areas change. This can be confusing so try to remember.
4. Use the Plot Options to adjust the histogram, by adding more bins and scaling the axis. If you have high or low outliers change the min or max values to see the general trends in the histogram.
5. Ensure you can see both the image and the histogram, and that the Active Subset selector (Fig 6) is set to 'Subset 1'. Then make a big change to the subset on the image and you will see the histogram change at the same time. This linkage is called 'brushing'.
6. Set the Active Subset selector (Fig 6) to 'None/Create new' then use region selection on the histogram to select a range of bins. You should now see a new subset in the Data area. Going back to the original image you should see the new subset highlighted across all the image, not just the original subset. You have created a new subset defined on all your original data.
7. When viewing the image, you can use the checkboxes, in the plot layers area, to select which subsets are visible.
8. Finally drag the original w5 data set to the plotting area and select histogram. You will now see a histogram of the whole image, with the histograms of any subsets superimposed in different colours. You may need to scale a lot to make the subsets big enough to see.

3. Linking

One of the strengths of GlueVis is the ability to view and cross link different data sources, including those that are not naturally visual. This is called linking.

1. For this we will use the data set "w5_psc.vot", open this while leaving the "w5.fits" loaded.
1. At the top click the "Link Data" button see the link editor pop-up. If the two data sets are not highlighted, then click on them to select.
2. The aim is to link attributes in the data sources that are the same, in this case the first pair of fields are **Right Ascension** in **w5.fits** and **RAJ2000** in **w5_psc.vot**, so select these in the scroll boxes. Then click the "Glue attributes" button to create a link.
3. The second pair of attributes are **Declination** and **DEJ2000**, so make a second link.
4. Finally click 'OK' to exit the Link Editor.
5. Shrink the existing window and drag the w5_psc data to the plotting area and create a 2D scatter (set (x,y) to (RAJ2000, DEJ2000)), then repeat but make a 3D scatter (set (x, y, z) to

(RAJ2000, DEJ2000, JMag)). You should see the previously defined subset in the new windows, see figure 7.

6. In the Data collection area click the arrow by your subset and you will see it contains a separate item w5_psc, drag and drop this onto the image to view the points. Use the point size tool (right click on the subset in the plot layers area) to adjust the size of the points.
7. Save these as a comma separated value (CSV) file points by right clicking on the w5_psc subset clicking "Export subset values". In the export dialog, figure 8, select w5_psc as data set, your subset as the subset and CSV as the file format.

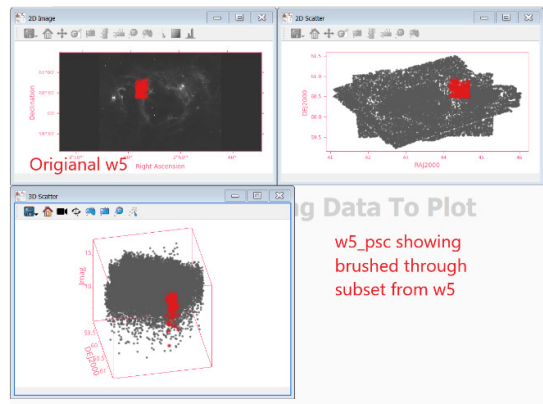


Figure 7 The original w5 data with a subset (top left) and two view of the linked data in w5_psc.

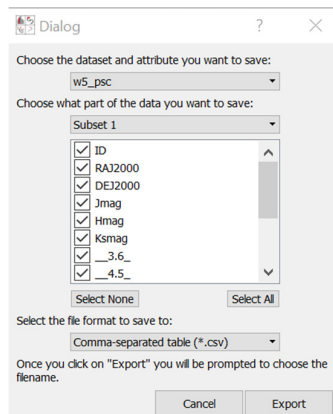


Figure 8 The top drop down selects the data set, the middle the subset, and the bottom the file format. In the scroll box you can check the columns of the table you want to output.

4. Adding New Attributes

In general data should not contain dependent fields, for example if you have a value for some property, you will not hold the normalized value as well. You calculate that when you need it, which avoids recalculation of every value if there is any change of the data. However now we may need to plot these so tool in GlueVis called the Arithmetic Editor and, for the maths, we will use a Python module called **numpy**, which is available in glue using the alias '**np**'.

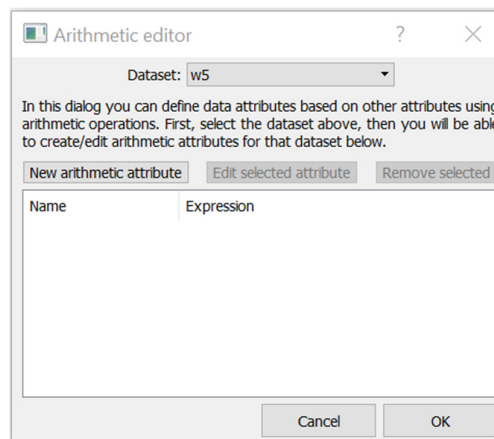


Figure 9. The Arithmetic Editor, the combo box at the top selects the data set on which to work, the "New Arithmetic attribute" button allows a new attribute to be created.

1. First view the raw data, drag "w5_psc" to the plotting area drop and select "Table view", to view the data. Note that some of the rows contain the error code for not a number "nan". Real data contains such errors, which they must be allowed for in the calculation.
2. From the top buttons click the one labelled "X² Arithmetic attributes" to pop-up the attributes box, see figure 9.
3. From the list at the top select "w5_psc" then click the "New Arithmetic Attribute" button.
4. In the equation editor add the name "Jnorm", then select "J mag" and click insert.
5. To normalize the data, we divide each value by the max value minus the min value, but if we use the Python functions max and min functions, we will get a errors because of the "nan" values. We avoid this by using the **numpy** `nanmax` and `nanmin` functions. So we enter:
$$\{jmag\} / (\text{np.nanmax}(\{jmag\}) - \text{np.nanmin}(\{jmag\}))$$
6. If "Valid expression" appears bottom left of the dialog, you can click "Ok", knowing your expression will execute. *It may not be correct, but it will run.*
7. In the table view there will now be a new column holding the normalized data.

5. 3D Data

For this exercise load the file earthquakes_2010_raw.csv, as this is plain text it may take some time.

1. Drag the data to the plotting area, and select “Table view”, you will see a database of earthquakes; note that the first two columns are latitude and longitude, while the last three are x, y, z, cartesian coordinates calculated from the latitude and longitude.
2. Drag and drop the data again, but select the “3D Scatter Plot”, you will see a cube containing data points, that are clearly not random.
3. The mouse controls are
 - a. Hold down left and drag: rotate whole plot.
 - b. Mouse wheel: scale plot in box.
 - c. Right and drag: magnification.
4. Look at the plot options panel, GlueViz has chosen the first three numeric columns for the x, y and z data, change these to the data’s x, y, z. You should see the plate boundaries of planet earth.
5. Look at the Plot Layers box, there are two tabs at the bottom (size and colour) which control the display of the data points.
6. On the “Size” tab change ‘Fixed’ to ‘linear’, this scales the data points according to a numerical column. From the drop down select mag, the magnitude of the earthquake.
7. On the “Color” tab and change to linear, select depth as the attribute by which to scale, then choose a colour scheme that you like.
8. Play with the sliders to see if you can identify a group of powerful and deep earthquakes. You can select and save this group.

6. Programming GlueViz

GlueViz can be extended by programming in Python.

6.1. The IPython Terminal

GlueViz has a built in [IPython terminal](#), which lets you access the currently loaded data, programmatically. Since GlueViz must handle data of many different types the data structures used to store the data are extremely abstract. This exercise, to average the magnitudes of earthquakes is intended to acquaint you with this abstraction.

IPython is a terminal combining interactive Python with a Unix/Linux terminal. As well as letting you type in Python command, you can use Bash terminal commands, the following are useful.

- **pwd**: name the present working directory
- **ls**: list contents of current directory
- **cd**: change directory

[Python3](#) provides several commands especially useful for interactive programming.

- **dir(), dir(<object name>):** returns list of the attributes and methods of any object , or the available in the local scope if called with no parameter.
- **globals():** return the current scope's global variables.
- **locals():** return the current scope's local variables.
- **help(<name>):** return the help string for the named object
- **str(<name>):** return a text string representation of the object, aimed at users.
- **repr(<name>):** return a text string representation of the object, aimed at programmers.
- **type(<name>):** returns the type of the object, such as list, function, float

6.2. Find & Manipulate Your Data

Internally the data you open is stored in a variable called **dc**, you can use this to access the data programmatically. In this exercise the objective is to visualize earthquake data and find earthquakes larger than magnitude 5.0.

1. Load the earthquakes data file (earthquakes_2010_raw.csv).
2. Drag into the plotting area, select 3D Scatter, then set the (x, y, z) to the data's columns (x, y, z). You should see the plate tectonic boundaries.
3. Open the IPython terminal (button labelled "Terminal", top-right).
4. In the terminal type `dir()` to list the what is available in the local scope. You should have an object listed called 'dc'.
5. Type `dir(dc)` to list the contents of dc (data_collection is the same thing).
6. The actual data is stored in the field 'data', type `dc.data` to list the data sets currently held by dc. The eqarthquakes file should appear.
7. To see how the earthquake data is stored, type `dir(dc.data[n])`, where n is the position (starting at zero) of earthquakes in the previous list, this will list all the fields in the object. To save typing assign the data to a variable by typing `eq_dat = dc.data[n]`, then type `eq_dat.components` to list the fields in the data.
8. You need to use the column names so type `columns = eq_data.components` this will save the column titles.
9. To get the component id for magnitude, which is seventh on the list, type `mags = eq_data.get_component(columns[5])`.
10. Type `mags.data` to check you have an array of numbers.

11. Now we need a means of selecting the large earthquakes so enter `selector = eq_data.id[columns[5]] > 5.0`. This will make a the selector for the subset, but it will not be evaluated yet.
12. To make the subset and give it a name enter `subset_group = dc.new_subset_group("big", selector)`, this will generate the subset and update the displays. Check you can see the subset in the scatter plot.
13. Note this is a subset group, another dictionary, so to see the actual subsets type `subset_group.subsets[0]`. To access the data `subset.subsets[0]['x']`.
14. If you left click the mouse in the terminal you can cut and paste the contents or save them as a web page.

8. Running GlueVis from a Script

If you must perform some analysis repeatedly you can write a simple Python program called a script, in which you preform the analysis then call GlueViz to display the results.

1. Put the file `earthquake_viewer.py` in the same directory as the earthquake data file.
2. Open a new command tool window, activate your glueviz environment and type `python earthquake_viewer.py earthquake_viewer.py`. It should pop up GlueVis with a 2D scatter plot of big earthquakes already selected.
3. Open the file `earthquake_viewer.py` in a text or code editor, and have a look, can you add a second 2D scatter of very deep earthquakes?

9. Adding Functions

For this exercise you need to know how to define functions in Python.

You can augment the functions available within GlueViz with your own additions. To do this you have to tell GlueViz where the functions have been located, this is done with a file called “`config.py`”, located in the directory “`.glue`” itself located in your home directory.

For this example we will add a link function. This is a function that allows the links to work with different data types

1. Start up GlueVis and load the `w5` and `w5_psc` data as in the first example.
2. Open the Link Data dialog, click the “Advanced” button, and check that the list of functions under the ‘General’ category does not include Celsius to Fahrenheit conversions.
3. Move the file called `demo_config.py` to the directory ‘`.glue`’ in your home directory. Now change the name to `config.py`.

4. Restart GlueViz reload the data and check the link functions again you should have Celsius to Fahrenheit conversions .
5. Open the config.py file with a text editor and examine it.

The file defines two python functions for converting temperatures. Above each function is the strange line

```
@link_function(info="Celcius to Fahrenheit", output_labels=['F'])
```

This is a decorator, a function which changes the nature of other functions. In this case it makes them appear to be part of the GlueViz link functions module. The info parameter is the title to appear in the user interface, and the output label tells GlueViz the unit's label.

References

[Hackable User Interfaces In Astronomy with Glue - aspbooks.org](#) Goodman, A.A. (2012). "Principles of high-dimensional data visualization in astronomy". *Astronomische Nachrichten*. **333** (5–6): 505–514. [arXiv:1205.4747](#). [doi:10.1002/asna.201211705](#). [ISSN 0004-6337](#).

GlueViz tutorials: <http://docs.glueviz.org/en/stable/installation/installation.html>

Python tutorial: <https://docs.python.org/3/tutorial/>

IPython tutorial: <https://ipython.readthedocs.io/en/stable/interactive/>

Introduction to python: <https://www.w3schools.com/python/>

Introduction to Numpy: <https://numpy.org/doc/stable/user/quickstart.html>

Introductin ot Scipy: https://www.w3schools.com/python/scipy/scipy_intro.php

Google Python Style: <https://google.github.io/styleguide/pyguide.html>