UNIVERSITY OF LEEDS

GlueViz
(glue)

J H Pickering

J.H.Pickering at leeds.ac.uk

University of Leeds

27 January 2022

Introductions:

I was originally a Materials Science, and worked as a research metallurgist, before taking a computer science conversion. After that I gained a PhD specializing in the application of artificial intelligence to computer graphics.

Since then, I was employed in various teaching, research and industrial roles, until I joined the University of Leeds. There I became a research assistant working on the deployment of molecular mechanics research software to industry. With the completion of that work I took up my current role in Software Engineering.

What is GlueVis?

- GlueViz is a package allowing heterogeneous data, spread across multiple tables to be loaded, linked, analyzed and visualized.
- Heterogeneous: items within the data have a diverse nature and are incommensurable because of their different natures.
- Heterogeneous data is common in the sciences:
    - Hourly temperatures and daily average rainfall
    - Average income and average school leaving age
- In many cases the data will be stored in separate files which have to be linked using one or more fields, hence the glue part of the name.
- It is often desirable to visualize the result in some way hence the viz part of the name.

Installing GlueViz

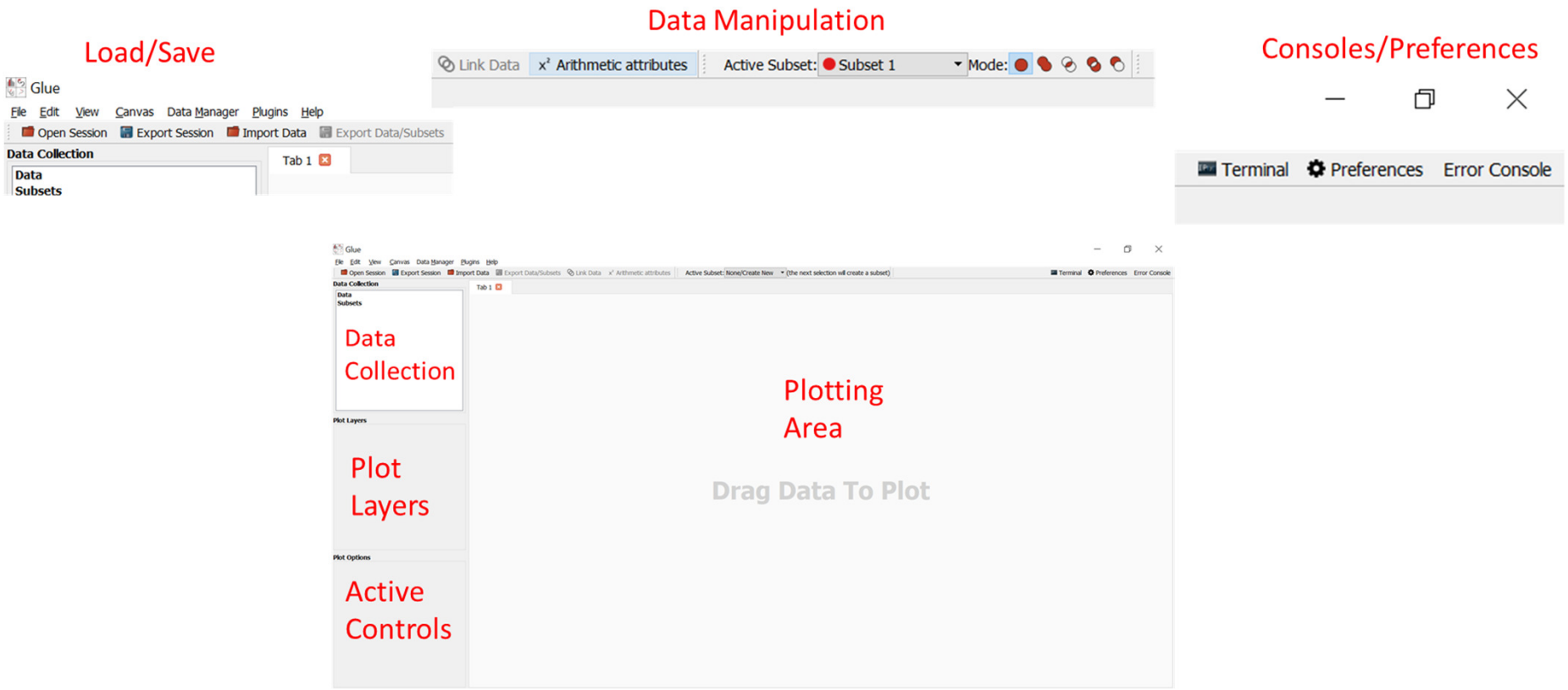
- The simplest way to get **GlueViz** is to install the **Anaconda** which is an open-source programming environment, bundled with multiple packages aimed at scientific computing.
- If you have Python and the package manager pip installed, you can get a copy via PyPi.
- Anaconda has its own package manager called **conda**, which installs the packages and keeps them up to date.
- Anaconda is available Windows, Linux and Mac, and can be installed as a user package (allowing the user to use **conda** to install new package) or as a system package (you will have to run as administrator to install packages).
- The packages in Anaconda may be older versions than are currently available, this is because they must all be cross platform integrated.

The advantages of programming GlueViz

Since glue was originally a Python module GlueViz has been equipped with an interactive Python IPython terminal. The glue libraries can also be called from within a Python program, and the results displayed for further analysis using GlueViz.

Programming allows all the features of glue and other Python libraries to be combined in many ways. Using the graphical interface limits the user to those that have been ‘canned’ and connected to buttons. The disadvantage is that programming is harder to learn than clicking buttons.

GlueViz: User Interface



Demo and Exercises

There now follows a quick demonstration of GlueViz, followd by the first set of exercises.

Python Programming Basics

Simple variable: the variable holds one thing

```
>>> X = 7
>>> Y = “the time of day”
```

Lists: the variable is a list of differ]rent things, indexed by the number of their location in the list.

```
>>> x = [7, “the”,]
>>> x.append(3.1415)
>>> Print(x[0])
7
>>> Print(x[1], x[2])
the 3.1415
```

Python Dictionaries

Dictionaries: the variable is a list of things, indexed by other things acting as keys

```
>>> x = {"blue": 7, 9: "data"}
>>> x.keys()
dict_keys(['blue', 9])
>>> x.values()
dict_values([7, 'data’])
x["blue"]
7
>>> x[78] = "hi"
>>> x
{'blue': 7, 9: 'data', 78: 'hi'}
```

Dictionaries are important because:

of the heterogeneous data GlueViz must handle, tables are read as dictionaries.

Internally glue stores tables in a DataCollection, available as the variable ‘dc’

```
>>> from glue.core import Data, DataCollection
>>> from glue.core.data_factories import load_data
>>> dc = DataCollection()
>>> dc.append(load_data("./w5_psc.vot"))
>>> star_catalog = dc.data[0]
>>> comps = star_catalog.components
>>> comps[0]
Pixel Axis 0 [x]
>>> star_catalog.data[comps[0]]
[    0,     1,     2, ..., 17768, 17769, 17770]
>>> star_catalog.data['Pixel Axis 0 [x]']
[    0,     1,     2, ..., 17768, 17769, 17770]
```

Three Commands Useful in interactive Python

type(x): returns the class of x (what kind of object x is)

```
x = 7
type(x)
<class 'int'>
```

dir(x): returns a list of attributes and methods in x

```
dir(x)
['__abs__', '__add__', '__and__', ….., 'imag', 'numerator',
'real', 'to_bytes']
```

help(x): returns the comments strings for x

```
help(x)
Help on int object:
class int(object)
 |  int([x]) -> integer
…….
```

Bibliography

- *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*

Thanks to:

- N8 High Performance Computing: https://www.n8research.org.uk/research-focus/legacy-programmes/n8-high-performance-computing/

- This work was funded by Joanna Leng's EPSRC funded RSE Fellowship (EP/R025819/1)

- The School of Computer Science, The University of Leeds, Leeds, UK