

ViewYt: A Push Towards Graphical Data Analysis

June 15, 2016

Austin Gilbert^{1 2}, Dr. Matthew Turk^{3 4}

1 Introduction

In addition to hair bands, a unified Berlin, and the first personal computers, the 1980's birthed the first computationally efficient methods to approximate the dynamics of n-body systems using large clusters of computers (*Hockney & Eastwood, 1988*). This opened the door to investigation of all sorts of complex phenomena from the dynamics of electrons in semi-conductors to the formation of the first stars and galaxies. However, with this capability came the need for problem specific analysis tools, which often came at the cost of fragmenting research collaborations and results that were not easily replicated. In the community of computational astrophysics, the YT Project⁵ (*Turk et al, 2011*) resolved these issues.

YT is a Python based package of tools for large scale data analysis and visualization of astrophysical simulations. It effectively enables users with a wide variety of data formats to analyze and present data in a common format. Additionally, it is open source and community maintained, enabling it to constantly keep up with the needs of its users. While it is a truly fantastic program, the majority of its functionality is based on tailor made scripts users apply to their data. This can slow down the initial review of data and it also prevents new users from quickly being able to utilize the power YT has to offer.

In light of this barrier, we intend to lay down the foundation necessary for the community to have a graphical user interface (gui) called ViewYt with all of the functionality yt has to offer, but with no required knowledge of programming to effectively utilize it. The primary goals of this gui are to be user friendly, intuitive, and computationally efficient. The rest of this proposal is structured as follows: in section two the generalized process of analysis on which this gui will be based is defined. In section three the packages necessary to implement this gui will be listed. Section 4 will discuss the stages of implementation and long term goals, and potential modifications to each. Section 5 will conclude this proposal and briefly review all previous sections.

¹Undergraduate, Georgia Institute of Technology, School of Physics

²2016 NCSA Spin Intern

³NCSA Researcher

⁴2016 NCSA Spin Mentor

⁵<http://yt-project.org/>

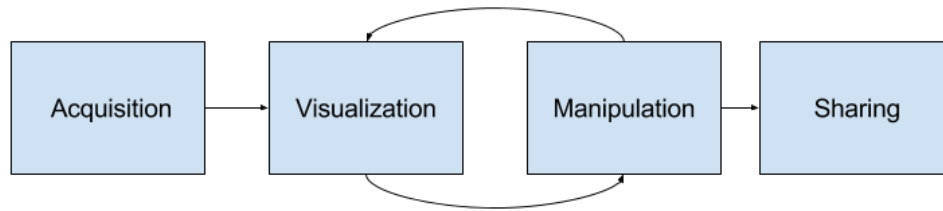


Figure 1: The fundamental process of analysis.

2 Data Analysis: A Broad Definition

At its core, analysis can be best represented by the path shown in figure 1. Analysis is unique in problem solving in that it can be essentially defined independent of the problem, however the precise implementation must often be tailored to problem at hand. In this case the primary problem is investigation of large scale physical phenomena, but this inherently brings with it innate problems related to handling large scale data and efficiently using available computational resources. Fortunately, YT largely resolves these problems, so all that is needed for a graphical implementation is to properly couple a visual display with YT that directs users through each stage of analysis while simultaneously enabling them to customize the display and the data at each step to suite their needs.

3 PyQt: The Prime Dependency

Since YT is based on Python, any gui implementation that could be decently coupled to the code would also need to be written in Python. For this reason we intend to use PyQt, a python wrapping of Qt(a C++ based package for visual interfaces), to create the gui frontend.

While PyQt is the main dependency on implementing a gui, things are rarely simple. In order to use PyQt in the gui, users will need a few other packages to be locally installed. First and foremost, users will need a means by which to install python packages, and options for this include PiP, Conda, and pulling directly from the source repositories of the individual packages. Additional packages that we expect will be needed are YT(obviously!), Qt4, Sips, Numpy, and Matplotlib, IPython, and H5py.

4 Stages Of Implementation

Before going on, it needs to be emphasized that development of this gui is not meant to end after the initial implementation. Instead the initial release is

meant to serve as the base off of which the YT community as a whole may build on as their needs arise. In light of this, the initial release will serve as the first stage of implementation. The intended features of such a release are as follows:

- Ability to load individual files from a simulation
- Ability to display all primary YT plot types (Slices, Projections, Phase Plots)
- Ability to annotate all primary YT plot types, and modify the spatial location being plotted through rotation, zoom, and panning.
- Capacity to directly interact with plots and gui through an IPython console

Beyond this stage, it is hard to document with certainty what other features ViewYt will have. However a few long term goals include:

- Ability to interact with remote datasets
- Increase the number of displayed plots based on user input
- enable 3d data visualization through an openGL widget and photorealistic rendering
- Linked displays for applying changes to a number of plots at once
- Handling of entire simulation outputs

Additionally, it should be mentioned that other Python packages might enable ViewYt to be realized far more rapidly. For this reason we are seriously investigating the possibility of integrating YT into glueviz, a Python package currently tailored for examining astrophysical data that is in the FITS format.

5 Review

With the rise of high powered computing and means to approximate complex phenomena has arisen a need for data analysis tools to investigate the resultant information. YT currently meets this need in the field of computational astrophysics, but it can be argued that it does not meet it to completion. New users, students, and those with no background in programming still have a moderate learning curve to climb before they can fully utilize YT. We intend to resolve this with ViewYT, a PyQt based graphical user interface built according to the primary steps of analysis. In addition to PyQT, ViewYT will require a number of other Python dependencies as its development continues into the future. Competent users in the YT community are free to make their own contributions to the project and assist with some of the large future goals we have in mind.

Once the gui is smoothly transitioned to the community as a whole, we can finally start doing some astrophysics too!