Proposal for Google Summer of Code 2020

Jupyter Notebook Widgets for TARDIS

By: Jaladh Singhal

1 Introduction

A GUI is very essential for TARDIS so as to allow researchers to easily investigate the important information about a simulated supernova model, without knowing its exact inner data structure. But the existing TARDIS GUI is outdated and is implemented in Qt which is no longer an ideal framework. These days, since most of the analysis work is done in Jupyter Notebooks, thus TARDIS GUI can also be brought to the notebooks by using Jupyter widgets and interactive plotting libraries.

Moreover, the existing GUI is limited in its functionality, providing only shell & line information. There is lot of other information that can be analysed from a simulation model which will be made accessible under this project, by means of new widgets. Besides, this project also aims to make it easier for users to try out these widgets themselves. For this, several example notebooks will be embedded in docs, *thereby showcasing the unprecedented possibilities with TARDIS!*

2 Project Objectives

Following is a breakdown of what I aim to achieve under this project:

2.1 Primary Objectives

2.1.1 Redevelop existing Qt GUI components as Jupyter Widgets

The existing Qt GUI consist of two interfaces for analysing shell info and for line info, both of which will be redeveloped for Jupyter Notebooks by tying tables & plots together with interactive widgets.

2.1.2 Develop new Widgets for other analysis tools of TARDIS

Other than the analysis tools for which Qt GUI exists, new widgets will be developed for several other code & tools that analyse or post-process the TARDIS models, like tardisanalysis tools.

2.1.3 Demonstrate the use of Widgets in TARDIS docs

Entire GUI guide in TARDIS docs will be replaced with multiple example notebooks, to show the use of each developed widget and how it can aid in analysing the simulation model.

2.2 Secondary Objective

Converting TARDIS docs into runnable notebooks

Majority of guides in the TARDIS docs describe its inner structure rather than showing what is possible with TARDIS. Thus, not only new notebooks demonstrating it will be added, but they will also be made runnable online, so that users can try it out themselves with ease.

3 Implementation Details

For accomplishing above objectives, the implementation of project will majorly involve following three parts (for sequential steps of implementation, refer timeline section):

3.1 Development of Widgets

To develop widgets for TARDIS GUI, I will primarily use following two tools:

1. IPyWidgets - Interactive HTML widgets for Jupyter Notebooks that is the core requirement of this project. It is also referred as Jupyter widgets.

2. Plotly.py - Open-source Python graphing library to make high-quality interactive plots.

Both of these play so nicely together that they make it possible to develop almost any complex graphical interface within Jupyter Notebook. IPyWidgets enables the UI controls (called widgets) to interactively update the Plotly plots as well as other widgets, which is demonstrated really well in this article. Plotly also provides an excellent support of IPyWidgets since each plot has charts events (like hover, zoom or click) which can be accessed by FigureWidgets.

I propose to use Plotly over other graphing libraries because it provides lot of interactive features out-of-box and is very intuitive to use, making development of a required feature faster. But if we encounter any limitations in implementing an interface, *this choice is flexible to changes*. Thus the first essential task of this project will be to **develop a prototype of the existing GUI** using ipywidgets & plotly. And if required, we can try implementing same prototype by using other graphing libraries like bqplot or bokeh, to see what works the best.

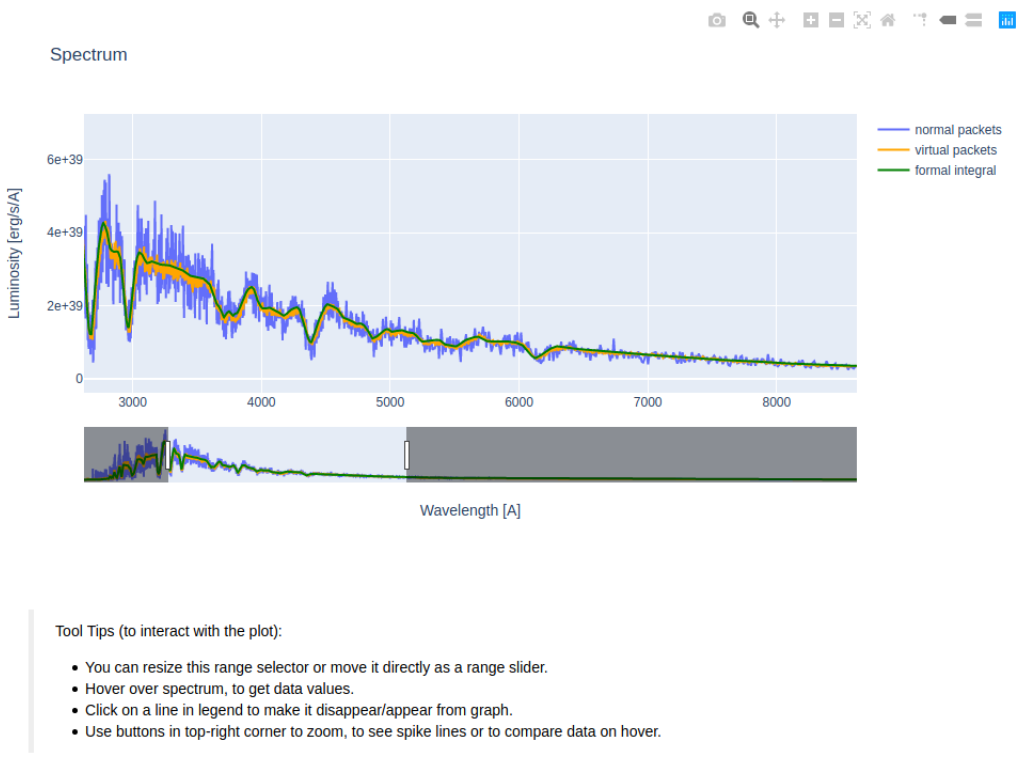

Figure 1: TARDIS spectrum (of quickstart guide) plotted using Plotly - check interactive version here

The developed widgets & plots will further be made into **more complex interfaces** by using the various layouts widgets. Since the output space in a notebook (where interface will be shown) is much less as compared to a full screen dashboard application, therefore I have planned to use nested tabs and accordions widgets for displaying multiple related components in one interface.

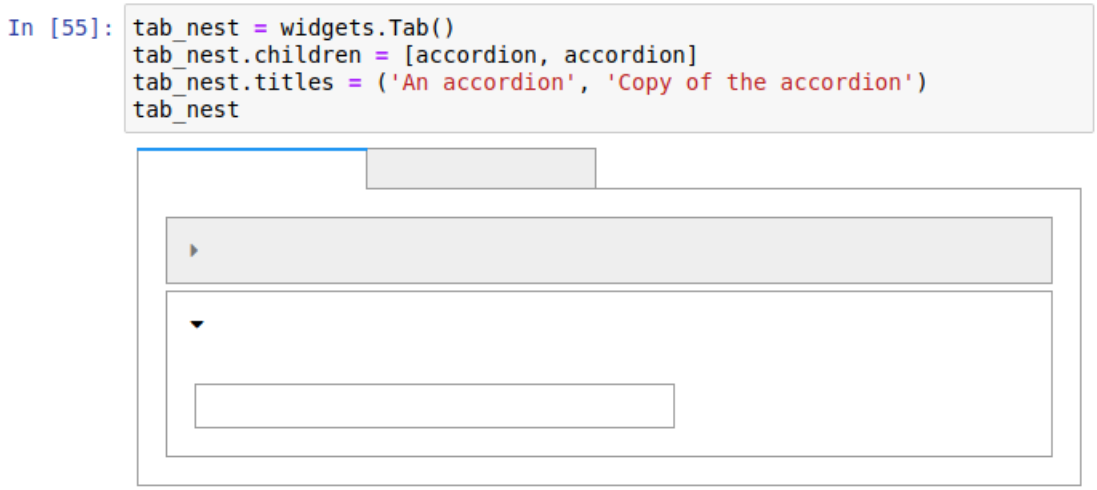

Figure 2: Use of accordions nested in tab widgets for complex interfaces

3.2 Development of GUI modules

Once an interface (or widget acting as GUI) will be developed in Jupyter Notebook, I will assemble its code from multiple notebook cells to write several functions as a part of GUI module. In this way, the widgets' code will become reusable & properly-organised and it will allow users to generate a required widget in their notebooks by a single function call. These high-level functions will only take **simulation object as a single optional argument** and default value of the argument will be a stored simulation object, to display widgets with minimum effort.

This process of writing functions in GUI modules will be iterative - followed after developing each interface, be it existing GUI components or new widgets. The structure of these new GUI modules can take inspiration from the existing GUI sub-package of TARDIS, where widgets and data handlers are kept in separate modules.

3.3 Enhancement of TARDIS docs by adding Notebooks

To show users what they can "do" with TARDIS, its docs require couple of enhancements. Firstly, the GUI guide will be completely replaced by example notebooks that demonstrate the use of each developed widget by calling functions from GUI module. Besides, I will also add the screen-casts of interaction with widgets to explain their working better.

Secondly, the notebooks in docs will be made runnable online, so that interested users can quickly try them within their browser, without any local installation. For this, we can use Google Colab, Binder or any suitable notebook running service. Other than quickstart and widgets notebooks, I aim to add more tutorial-style notebooks and list them together in an index notebook - similar to this IPyWidget's collection of example notebooks that can be instantly run by a "launch binder" button in their readme.

4 Timeline

4.1 Weekly Timeline

During the three-month period of Summers, I will devote 35+ hours per week to this project. The tentative timeline of what I have planned to accomplish each week, is as follows:

- **Week 1:** Develop prototypes of shell & line information interface within Jupyter notebooks, using IPyWidgets & Plotly and other tools (if needed).
- **Week 2:** Choose the best prototype and complete it into a well-finished interface.
- **Week 3:** Discuss structure of GUI modules and assemble the code of interfaces in notebook as the functions in modules.
- **Week 4:** Finish GUI modules and prepare notebooks that display the interfaces by calling functions from those modules.
  Deliverable produced: *Jupyter widgets for existing GUI components*
- **Week 5**[1]**:** Discuss other analysis tools of TARDIS for which new widgets can be developed.
- **Week 6:** Design wireframes and develop prototypes for the interfaces of chosen analysis tools.
- **Weeks 7-8:** Develop widgets for the interfaces, add new functions to GUI modules and prepare notebooks using them.
  Deliverable produced: *New widgets for other analysis tools of TARDIS*
- **Week 9**[1]**:** Add more documentation in prepared notebooks to demonstrate the use of widgets better.
- **Week 10:** Put all example notebooks together in GUI guide of TARDS docs along with suitable screen-casts.
  Deliverable produced: *Example notebooks in docs demonstrating the use of widgets*
- **Week 11:** Discuss the choice of tool to make the notebooks in TARDIS docs, runnable online and add notebooks for other guides if necessary.
- **Week 12:** Other improvements in TARDIS docs.
  Deliverable produced: *TARDIS docs as runnable notebooks*

4.2 Other Commitments

Most likely, I will be having my semester exams from mid-April to mid-May which will be during the community bonding period. So it will have negligible affect on my progress on the project, since I am already familiar with TARDIS community. Other than that, I have no other commitments planned as of now.

[1] Evaluation Week

5 About me

5.1 Personal Information

- **Name:** Jaladh Singhal
- **E-mail Address:** jaladhsinghal@gmail.com
- **GitHub Username:** jaladh-singhal
- **Location:** Jaipur, Rajasthan, India
- **Time Zone:** Indian Standard Time (UTC+5:30)

5.2 Past Contributions

I am engaging with TARDIS community since last year when I got selected as a GSoC'19 student in its sister organisation - StarKit. Until now, I have attended several weekly hackathons of TARDIS, interacted with superb Astronomers & Computer Scientists and tried to help them where I could. I have also reviewed multiple PRs of the potential students for GSoC'20 and clarified many of their doubts on Gitter channel, as an active member of TARDIS.

As a part of this project's first objective, I have opened PR#1107 in which I have plotted spectrum (of quickstart guide) and implemented several components of shell information Qt-GUI, using interactive features of Plotly.

5.3 Background Information

I am a Computer Science Junior who has a keen interest in Research Software Development for Astrophysics. Working on an Astrophysics software, keeps piquing my curiosity because it provides me opportunities to understand not only the software components but also the underlying concepts of Astronomy. Getting selected in GSoC'19 with StarKit, was the product of pursuing this interest along with my continuous efforts to learn and apply new knowledge. I am also a well-experienced Pythonista, who has become an Open source enthusiast after realizing the power of communities since last year.

In GSoC'19, a part of my project was to develop interactive web apps for StarKit to make the analysis process easier & faster for users. That's why, I possess considerable knowledge about interactive plotting and graphical interface development which will be fruitful for this Jupyter Widget project as it goes along the same lines. I always get excited by the strength of Information Visualization in simplifying complex things and making users engaged. Therefore, I am pumped up to work on this project with TARDIS in coming Summers, which is at the crossroads of both Astrophysics and Information Visualization.