Software Development within STScI's TIKE Server to Increase Accessibility of TESS Exoplanet Transit Data to Introductory Astronomy Students

Templeton, Alexander
Advisor: Dr. Edward Murphy

Product downloadable via Github: https://github.com/ajttja/TIKE-Exoplanet-Transits
Product runnable only via the TIKE server: https://timeseries.science.stsci.edu/hub/spawn

Introduction

A sorely underutilized tool in teaching astronomy concepts to new students of astronomy are exoplanet transits. They have the potential to be easy to understand, while demonstrating the concepts of light curves, creating time-series data from a single pixel, how Kepler's laws can be practically used, and generally how astronomers are able to imply a great deal of information from relatively little data. It also allows students to leave the realm of textbooks and the theoretical and actually work with real data so that astronomical science might feel more accessible.

In the past, one hurdle in making exoplanet transit data analysis accessible to beginners has been the difficulty of actually finding, downloading, and storing processed data. TESS produced around 7 TB of data a month, and storing all that data just to find a few good examples of exoplanet transits would be impractical. Recently, however, the Space Telescope Science Institute (STScI) has created the Time Series Integrated Knowledge Engine (TIKE). TIKE is a virtual server open to the public to use from which to easily access all the data from the Mikulski Archive for Space Telescopes (MAST), as well as all the software needed to analyze it (though with some limitations noted later).

So while there now exists an incredible resource for improving the accessibility of astronomical data, there remains a gap between the data and amateur users which is data interpretation. Bridging this gap is thus the primary focus of this project. Creating a program to make use of TIKE accessible data to allow students with little to no prior coding or astronomy experience to learn how to analyze exoplanet transit data, and then actually do so with real data from the Transiting Exoplanet Survey Satellite (TESS).

Development

As a baseline, I was working from a concept lab created by Edward Murphy. While this version would have served well for those already familiar with how to use python and an intuitive understanding of light curves, neither of those assumptions would have necessarily held for the target audience of those new to astronomy. Thus, there were immediately several potential improvements for increasing the accessibility of the lab. The smaller one would simply be to clean up the graphs and further integrate them into the explanations of concepts to rely less on written explanations. The larger and more difficult improvement would be to hide all the code and eliminate the need for the user to understand python at all. To achieve this while not sacrificing the adaptability of modifiable code, all the interactivity would need to be moved to the graphs themselves and adding an easy to understand user interface to them. Adding interactivity to the graphs would have the added bonus of allowing the users to experience for themselves various cause and effect relationships which would hopefully further their understanding of how exoplanet transit data relates to what is actually happening in the real world.

The first approach taken towards both of these needed improvements was to use the Bokeh library. This library contains tools to make professional looking plots with a wide variety of interactive UI options. Indeed, with very little effort, the transit data was able to be presented in a highly professional manner, satisfying the first needed improvement. The issue with using Bokeh arose in the interaction between Bokeh's method of interactivity and the nature of the TIKE server. Bokeh manages interactivity by opening a Bokeh server to keep track of modifiable variables. Whenever an interactive element is used, the changed variable is sent to the server, and the server returns the updated plot to the main program. The default way of achieving this is for the program to open an internet connected browser in which to display the various plots and interactive elements. Alternatively, the plots can be displayed within the Jupyter Notebook itself, but the bokeh server needs to be manually opened via the terminal, or via commands within the notebook connecting to the terminal.

In its current configuration, the TIKE server has no ability to open up a browser, to use the terminal, and likely has limited outside internet access, and thus is unable to use a bokeh server no matter the method attempted.

To circumvent this limitation, instead of using Bokeh, I used ipywidgets for all the interactive UI elements, along with matplotlib for the actual plots. I attempted to use ipywidgets to add interactivity to the nicer looking bokeh plots without needing to use the bokeh servers, but unfortunately ipywidgets only has compatibility with matplotlib.

Final Product

Refer to the links at the start of this document. An account needs to be created to access the TIKE server, but this is a quick and easy process.

Potential Future Improvements

Despite the inability to use the Bokeh library in this project for the reasons outlined in the Development section, its potential abilities remain as clearly superior to those of ipywidgets and matplotlib. Were the TIKE server to add the ability to use Bokeh servers, or the data to be compiled into an easily downloadable folder that reduced the need to use the TIKE server to begin with, then the plot elements of the lab should be replaced with bokeh plots. Additionally, the added functionality of drop down menus provided by Bokeh would mean all the data used could be pre-loaded from the start and users could swap between exoplanet light curves by simply selecting a different star, eliminating the last line of code users have to interact with.

Rather than only having a preset selection of exoplanet transit data to examine, the code could be altered to either take in the name of any star with data, or produce a random dataset. The issue is that the vast majority of datasets contain no transit. Even those with confirmed transits are often too noisy to be clearly seen without advanced statistical modeling and would likely just confuse users.

Though understanding advanced statistical modeling is beyond the math ability assumed for the target audience, it could be possible to average data points that are folded on top of each other in order to limit the transit to noise ratio. The issue here is that for the cases in which this method is necessary, the data would be so noisy that the folded light curve would not show any useful information unless the period were already exactly correct, and with no clear transits beforehand to make a guess at the period, finding it would require users to very slowly check every period in the folded light curve. However, this would allow users to analyze data for significantly smaller exoplanets, and provide an understandable example of how useful information can still be extracted from noisy data.

Conclusion

Though the lab created was successful in its own right, the process of creating it also revealed two large opportunities for increasing public accessible astronomy. The first is the potential use of Bokeh as a tool for presenting research results in a way understandable for those outside the astronomy community. The second is to further make use of the TIKE server for similar projects to this one, but for datasets from other telescopes within MAST to explain concepts beyond just exoplanet transits. Whichever the approach, there is clear potential for progress in virtual astronomy public outreach.