Jupyter Notebooks for Enabling Non-Programmers Conduct Big Data Visual Analytics

Suresh K. Bhavnani PhD¹, Raed Zuhour MD², Tianlong Chen PhD¹

¹Institute for Translational Sciences, ²Internal Medicine, Radiation Oncology, University of Texas Medical Branch

Introduction

The explosion of digital data, coupled with powerful open-source software, has provided unprecedented opportunities for data scientists to accelerate scientific discoveries. More recently, this opportunity is being exploited through the use of Jupyter Notebooks,¹ which have enabled complex software pipelines to be more accessible, understandable, and reusable by analysts other than the original developers. However, while these data resources and technologies have been exploited by analysts with a background in science, technology, engineering, or math (STEM), they can be a hurdle for non-STEM disciplines such as nursing and occupational therapy. Users from such disciplines could greatly benefit from opportunities to leverage powerful analytical tools, without the need of being proficient in coding. Here we demonstrate two Jupyter Notebooks that were developed and iteratively refined with feedback from non-programmer medical students attending a course on big data visual analytics.

Method

Jupyter (a portmanteau of the software languages Julia, Python, and R) are editable documents that enable the combination of live code (e.g., R code that opens a specified datafile and displays the number of rows), narrative text (e.g., description and instructions on how to use the code), and graphics (e.g., visuals on what to expect as an output). The document is run through an application (e.g., Anaconda that runs the live code through a “kernel”), which can be executed on a personal laptop, or on a server. The Jupyter Notebook content can be structured in “cells”, which are independent executable and viewable sections for each of the above types of information, or can be structured as a batch process on a server where the code is run without user interaction. Jupyter Notebooks can be shared on open-source platforms such as GitHub, and are increasingly being required by funding agencies (e.g., American Heart Association) for sharing analytical pipelines.

We used the Jupyter framework to develop two notebooks that were designed for students with minimal to no programming background: (1) Bicluster Identification and Data Transformation. This notebook was developed for enabling the automatic identification of biclusters in large datasets through the use of modularity, and formatting the output for analysis by a sever-based application called ExplodeLayout.² The targeted application used machine learning methods to optimally separate and display biclusters in highly-dense bipartite networks, facilitating their interpretation and application in precision medicine. (2) Server-Based Processing to Measure Significance. This notebook was designed to measure the significance of the biclustering conducted by modularity. As this computationally-expensive method requires 1000 random permutations of the data, the notebook specifies how to conduct the analysis through the use of parallel multiple cores. Both notebooks were developed by a programmer, and iteratively refined after a demonstration by the instructor, and subsequently after use by the students to analyze their own data.

Results and Demonstration

The notebooks underwent three types of improvements, ultimately enabling students to use them independently: (1) Declarative Knowledge. Both notebooks assumed knowledge of concepts that confused some students. For example, most Windows/Mac users are familiar with the concept of “folder”, but not of “directory” which confused users when transferring files to the server. (2) Procedural Structure. Although the programmer had correctly described the steps required to run the program (Fig. 1A), it lacked procedural structure for comprehending the steps and preventing out-of-sequence execution¹ common in the use of such notebooks. This was addressed by having a common procedural structure: goal, data input format, method, and output (Fig. 1B). (3) Redundant Graphical Instructions. For critical conceptual and procedural knowledge, the students suggested including graphics even if it was redundant with the text, to provide visual orientation in the overall pipeline.

While the above two Jupyter notebook demonstrations are specific to big data visual analytics, the lessons learned from refining them should enable the design of other notebooks targeted towards motivating non-programmers to analyze their own data, and engage more fully in the data science revolution.

References


Fig. 1. A Jupyter notebook designed by a programmer (A), and iteratively refined through input from non-programmers (B).

1001