Current Problems in Diagnostic Radiology ▮ (2017) ▮▮▮–▮▮▮

Current Problems in Diagnostic Radiology

journal homepage: www.cpdrjournal.com

Scientific Notebook Software: Applications for Academic Radiology

Michael L. Richardson, MD[a,*], Behrang Amini, MD, PhD[b]

[a] Department of Radiology, University of Washington, Seattle, WA
[b] Department of Radiology, M.D. Anderson Cancer Center, Houston, TX

Objectives: The goal of this article is to introduce the concept of scientific notebook software, and to illustrate how it can be used to document a research project, to perform image analysis and to create interactive teaching tools.
Methods: We describe the installation of scientific notebook software known as Project Jupyter, which is free, open-source and available for the Macintosh, Windows and Linux operating systems.
Results: We have created 2 scientific notebooks that demonstrate applications germane to radiologists, particularly those in academic radiology. The first notebook provides a tutorial that summarizes basic features of the Project Jupyter notebook, and gives numerous examples of how the notebooks can display explicatory text, perform statistical computations, and display plots, interactive graphics, and audio files. The second notebook provides a toolbox for viewing and manipulating images in the Digital Imaging and Communications in Medicine format.
Conclusions: Scientific notebook software allows its users to document their work in a form that combines text, graphics, images, data, interactive calculations and image analysis within a single document. Scientific notebooks also provide interactive teaching tools, which are can help explain complex topics in imaging physics to residents.

Introduction

Interactive notebook software is experiencing a rise in popularity in the scientific and engineering communities,[1,2] and deserves strong consideration by radiologists, particularly those in academic radiology. This software offers its users an environment for exploration, collaboration, and visualization.[3]

Scientific notebook software has been around for some time. Some of the first notebooks were available in proprietary packages such as Mathematica,[4] Maple[5] and Matlab.[6] More recently notebooks have gained traction in the Python community with development of the iPython Notebook (now Project Jupyter).[7] Today there are many other notebooks to choose from, including RStudio,[8] Apache Zeppelinm,[9] Spark Notebook,[10] Databricks Cloud,[11] and Sage MathCloud.[12] These notebooks provide kernels (programming languages) for multiple popular languages, such as Python, Julia, Scala, SQL, and others.[3]

Scientific notebooks make it easy to share one's work with collaborators and other investigators via email or the web. Being able to access all of a project from within a notebook makes it easy to reproduce any part of this work, either by the investigators themselves or by other workers. These notebooks are also finding increasing use in presentations or interactive teaching sessions, where they are replacing PowerPoint and other static presentation tools.[13,14]

The purpose of this article is to introduce the concept of scientific notebook software, and to demonstrate how it can be used to document a research project or to create an interactive teaching tool. The first part of the article introduces the basic features of the notebook software, and gives a number of interactive examples that are germane to academic radiology. The second part of the article provides a toolbox for viewing and manipulating images in the Digital Imaging and Communications in Medicine (DICOM) format.

Methods

In this article, we focus on the scientific notebook software known as Project Jupyter,[7] which is free, open-source and available for the Macintosh, Windows and Linux operating systems. Jupyter has its roots in iPython (the "i" is for interactive), which was first developed in 2011. In 2014, the notebook and other parts of the project were spun off into Project Jupyter, which aims to make iPython more compatible with other languages. The name Jupyter is actually a portmanteau of Julia, Python and R, 3 of the first languages supported by this project. We did not write Jupyter, but are enthusiastic users of the software.

Using Jupyter

To use Jupyter, one must access a Jupyter server using a web browser. This Jupyter server can be located remotely somewhere on the internet, or locally, on one's own computer.

Portions of this work were presented at the Society of Academic Bone Radiologists 2017 annual meeting, Gloucester, MA.

* Reprint request: Michael L. Richardson, MD, Department of Radiology, University of Washington, 4245 Roosevelt Way NE, Seattle, WA 98105.
E-mail address: mrich@uw.edu (M.L. Richardson).

http://dx.doi.org/10.1067/j.cpradiol.2017.09.005
0363-0188/

The Try Jupyter Temporary Notebook Service

Project Jupyter has set up a temporary notebook service that allows one to test-drive Jupyter without having to first install and configure it on one's own computer. One can launch a temporary Jupyter notebook via the following steps:

1. Point a web browser to the Try Jupyter site at https://try.jupyter.org.
2. Choose one of the demonstrations for different programming languages, such as R, Julia or Python. For example, one might select the "Welcome to Python.ipynb" demo.
3. One can then select cells in that tutorial by clicking on them.
4. After selecting a cell, one can evaluate or run that cell by pressing shift-enter on one's keyboard.
5. The example code on the site will generate and plot a random set of simulated stock market data.
6. You are welcome to click the "+" symbol on the toolbar at the top of the Jupyter window to create new cells and enter and evaluate your own text or Python expressions.
7. This temporary notebook will disappear after 10 minutes of inactivity.

Installing Jupyter on Your Own Local Computer

The optimal way to use Jupyter is install and run it on one's own computer. This is a relatively simple process if chooses one of the following systems:

1. Enthought Canopy (https://www.enthought.com)
2. Anaconda Python (https://www.continuum.io/)

Both the Enthought and the Anaconda systems offer free versions. Both systems install not only the Jupyter notebook software, but also a complete distribution of the scientific Python (SciPy) and numerical Python (NumPy) libraries. Of the 2 systems, we currently prefer the Anaconda package, which we have found to be easier to install and update, especially for newcomers to Jupyter. Once installed, running the Anaconda Navigator app brings up a home screen with multiple modules (Fig 1). Clicking the "Launch" button in the Jupyter notebook pane will start a Jupyter session.

Results

We have created 2 scientific notebooks that demonstrate applications germane to academic radiologists. The first notebook provides a tutorial that summarizes basic features of the Project Jupyter notebook, and gives numerous examples of how the notebooks can display explicatory text, perform statistical computations, and display plots, interactive graphics, and audio files. The second notebook provides a toolbox for viewing and manipulating images in the DICOM format.

The ideal way to view our notebooks is in their full interactive form as a Jupyter notebook. However, for those who have not yet installed Jupyter on their computer, we have also supplied non–interactive versions of these notebooks as static web files.

Notebook 1: Jupyter Tutorial

We have created a multi-part tutorial file called "Jupyter_tutorial.ipynb", which can be downloaded from http://uwmsk.org/jupyter/Jupyter_tutorial.ipynb.zip. A non–interactive version of this tutorial is also visible as a static HTML file at: http://uwmsk.org/jupyter/Jupyter_tutorial.html. If you are test-driving a Jupyter notebook on the *Try Jupyter* site, you can copy and paste pieces from this tutorial into that notebook. Once you have installed Jupyter on your own local computer, you can open and run the full notebook there. We have successfully run all of these examples on both the Anaconda and Enthought systems. All of the examples included were written with Python 3, and some of them may not run properly under Python 2.

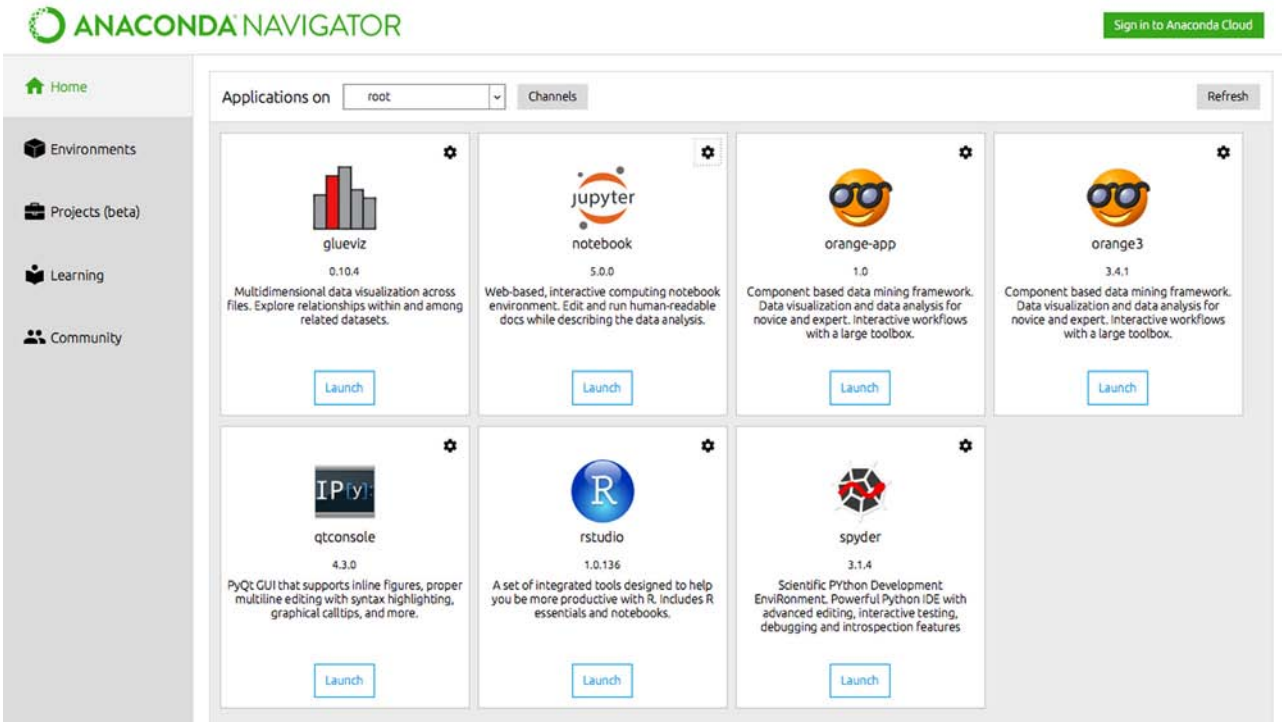

FIG 1. The Anaconda Navigator home screen on a Macintosh. (Color version of figure is available online.)