webinar_1-introducing_jupyter

January 22, 2019

1 Computational skills for Biocuration

2 Webinar 1

- Date: January 22, 2019
- Time: 16:00 GMT
- Connection Information: https://attendee.gototraining.com/r/2734438535500774913 (Training ID: 461-010-428)
- Notepad: https://piratenpad.de/p/2019-Biocuration-Module-2

2.1 Introduction to the course and platform

2.1.1 Who we are?

(listed in an order of the webinar content they will cover today)

- Malvika Sharan (email: malvika.sharan@embl.de)
- Toby Hodges (email: toby.hodges@embl.de)
- Marc Gouw (email: marc.gouw@embl.de)

2.1.2 Aim of the course

- To provide participants with a foundation in Python programming language for working with bioinformatics tools
- To provide a working knowledge of database structure
- To apply these skills to increase the efficiency of a set of curation tasks

This course will comprise of the following events

- Webinars and in-person training

Date	Event Type
Jan 22	Introduction (session will be recorded)
Jan 28	Troubleshooting/Q&A (session will be recorded)
4 & 5 Feb	**In-Person Training @EMBL-EBI**
11 Feb	Post-training session 1

Date	Event Type
18 Feb	Post-training session 2
25 Feb	Post-training session 3
4 Mar	Post-training session 4
11 Mar	Post-training session 5

- Course materials will be available at GitHub:https://github.com/zencore/2019-Biocuration-Module-2

2.1.3 Learning outcomes

- Discuss basic programming concepts in Python
- Create basic scripts
- Application of programming skills in the process of curation
  - Communicate with and query a data repository
  - Analyse and visualise biological data

Additional learning outcomes

- Understand and describe the structure of an appropriate data repository
- Describe key issues in producing both external and internal interfaces
- Explain the importance of discussion with interested parties eg software developers and database managers

2.1.4 Final assignment

Students will be assessed summatively by the completion of a computational based task, which will include the writing of code to undertake a specific curation task (**more about it at the in-person training**).

2.2 Python Programming

Python programming language is comparatively simpler to understand for newer learners. Though easy to read (due to it's simpler syntax), Python is powerful, making it an ideal language for the beginners to learn.

You can do really cool things in Python, starting from basic computing to high-throughput data analysis (visualization, statistics), machine learning, and creating reproducible workflows.

It is common to write Python codes using editors (gedit, spyder, notebook++ etc.) and save them in a plain text file with an extension *.py*. However, for this course we will use the Jupyter Notebook.

Jupyter Notebook In a Jupyter *Notebook*, we can do interactive programming (e.g. Python), by writing new codes or testing prewritten programs while storing them with text/notes for documentation.

- Jupyter is a set of tools that allows us to use Python and data while displaying the outputs and figures on the same browser.

- Like other specialized editors, Notebook provides code completion and other helpful features to write Python (and a few other programming) language efficiently.
- Notebook files have the extension .ipynb (stands for IPython notebook - previous name for Jupyter Notebook) to distinguish them from plain-text Python programs.
- You can type, edit, and copy and paste blocks of code in the "code cell".
- You can write notes and documentation in the "text cell".
- .ipynb files can be shared with others who can open them in their browsers, and see your work while reproducing results for your analysis.
- You can also generate/export a .pdf files with the output/images produced by your codes along with the notes (think about publications) and share with your collaborators.

Using Jupyter Notebook in your computer To use Jupyter notebook, you must install anaconda for your opertaing system along with the latest Python packages. The [Anaconda package manager]|(https://docs.continuum.io/anaconda/install) is an automated way to install the Jupyter notebook. - See the setup instructions for Anaconda installation instructions.

Once you have installed Python and the Jupyter Notebook requirements, open a shell and type:

```
$ jupyter notebook
```

This will start a Jupyter Notebook server in your computer (without internet) and open your default web browser. (The server does the work and the web browser renders the notebook)

Creating and storing Jupyter Notebook You can start a new page by clicking the button 'new'-> 'Python 3'.

You can test run a `print` command, to print something:

- Type the code written in the "code cell" below.
- Run by Shift+Enter or the play symbol (right arrow) in the toolbar on the top of your page.

```
In [11]: print("testing Python in Jupyter Notebook")
```

```
testing Python in Jupyter Notebook
```

A few more useful Jupyter Notebook tips To create new cells in Jupyter Notebook, you can use the following keyboard shortcuts: - New cell below: Esc+b - New cell above: Esc+a - Delete your current cell: Esc + X - Execute the contents of your current cell: Shift+Enter - Undo your last cell deletion: Esc + Z - See list of all the shortcut keys: Esc + H - Define type your current cell as "text cell": Esc + M - Define type your current cell as "code cell": Esc + Y

Markdown for documentation/notes

> Markdown is a lightweight markup language with plain text formatting syntax. - Wikipedia

Markdown is used for creating rich text using plain text. Markdown does most of what HTML does.

Text
Line breaks don't matter.
But blank lines create new paragraphs.
Headings

3 Level-1 Heading: Starts with '# ' (hash symbol and a space)

3.1 Level-2 Heading: Starts with '## '

3.1.1 and so on...

Exercise:

Upto which level of heading can you go?
Unordered List

- not numbered: using "-" or "*" or "+"
  - indented: using indentation and then "-" or "*" or "+"

Numbered list

1. First point: Starts with "1."
2. Second point: Also starts with "1."
3. Number don't matter: Also starts with "1."
   1. Sub-point: indented and then starts with "1."

Highlighting

- *italics*: Starts and ends with an underscore"_"
- **bold**: Starts and ends with two underscores "__"

Exercise

How will you generate text that are both ***bold and italics***?
Creating link
Markdown Cheatsheet - Text to be displayed in a square brackets **[Markdown Cheatsheet]** followed by a link in a round brackets (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) - The closing of square bracket and opening of round brackets don't have any space in between, i.e. "]("
That's all for now! Let's move on to Python programming.

4 Beginning with Python

Here we will introduce you to the basic Python commands and syntax.

```
In [12]: ## Let's try a few Python commands
```

Homework Introduction to Python Programming

1. Download the file anywhere in your computer
2. open a shell and type: jupyter notebook
3. This will start a Jupyter Notebook server and open your default web browser from where you can access the dowloaded file and open the rendered version on your browser
4. Read through the material while writing, running and/or testing Python codes
5. Do the exercises and reach out to us with any particular questions about the given homework

We will see you on at our second webinar.

- Date: January 28
- Time: 16:00 GMT
- Notepad: https://piratenpad.de/p/2019-Biocuration-Module-2

More information will follow in an email!

References and Recommendations

1. For more on Jupyter and markdown: Running and Quitting, Software Carpentry
2. Markdown Cheatsheet
3. Jupyter Tips & Tricks
4. Python wiki for non-programmers
5. Beginners guide for people with basic programming experiences