Python for scientists

Python, Spyder, Jupyter and Anaconda

def complementary(seq):
    nt_comp = {
        'A': 'T',
        'C': 'G',
        'G': 'C',
        'T': 'A',
    }
    compseq = ''.join([nt_comp[nt] for nt in seq])
    return compseq

Alvaro Sebastián Yagüe

www.sixthresearcher.com

@SixthResearcher
Python and Python Packages
Python

Python is a programming language in which we write computer programs called scripts. These scripts would be stored in text files that have the ending ‘.py’, for example ‘hello.py’.

Python programs are executed by an interpreter, on windows it is called ‘python.exe’ and from a command window we could execute the ‘hello.py’ program by typing:

> python.exe hello.py

On Linux and OS X operating systems, the Python interpreter program is called ‘python’, so we can run the program ‘hello.py’ as:

> python hello.py

(This also would work on Windows as the operating system does not need the ‘.exe’ extension.)

Python packages

For scientific computing and computational modelling, we need additional collections of Python modules called libraries or packages. They are not part of the Python standard distribution. These allow us, for example, to create plots, operate on matrices, and use advanced numerical methods.

The packages we generally need in science are:

- **NumPy** (NUMeric Python): matrices and linear algebra
- **SciPy** (SCIentific Python): many numerical routines
- **Matplotlib** (PLOTting LIBrary): creating plots of data

You can check the full list of Python packages in the [Python Package Index](http://www.southampton.ac.uk/~fangohr/blog/installation-of-python-spyder-numpy-sympy-scipy-pytest-matplotlib-via-anaconda.html) (currently 96220).
Spyder

Spyder is a powerful IDE (interactive development environment) for the Python language with advanced editing, interactive testing, debugging and introspection features.

The name SPyDER derives from "Scientific Python Development EnviRonment" (SPYDER).

Useful features include:

- **An advanced editor** with syntax highlighting and code completion help.
- **A console** to run and test the scripts.
- **A debugger** to find errors in the code running it line by line and checking the content of the variables.
Spyder

Editor

Debugger

Console

Alvaro Sebastián Yagüe

www.sixthresearcher.com

@SixthResearcher
Jupyter Notebook
Jupyter Notebook (former IPython)

The Jupyter Notebook is an interactive programming environment, in which you can combine Python code and execution results with annotations, equations, figures, links, etc.

The Jupyter Notebook will run in the internet browser of our computer and does not require internet access, it will access to our local files via the browser interface.

Notebooks are similar to Word documents and we can edit them interactively in the internet browser. Notebooks are saved in its own format with the ‘.ipynb’ extension, but can be also downloaded as PDFs, HTML pages or Python code.
Jupyter Notebook

Simple spectral analysis
An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n \exp \left( \frac{-2\pi i kn}{N} \right)$$

```python
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's built-in specgram routine:

```python
fig, (ax1, ax2) = plt.subplots(1,2,figsize=(16,5))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```
Anaconda
**Anaconda**

Anaconda is a Python distribution. A Python distribution provides the Python interpreter, together with a set of Python packages and sometimes other related tools, such as editors.

Previously we were using the official Python distribution nicknamed CPython. You can check an expanded list of Python distributions in this link.

The main advantages of Anaconda distribution is that it includes:

- NumPy, SciPy, Matplotlib and Biopython
- Spyder
- Jupyter Notebook

Here is the full list of packages included by the Anaconda distribution.
Downloading and installing Anaconda

Download Anaconda with Python 3 version (32 or 64 bits) from here:
https://www.continuum.io/downloads

Execute the installer and follow the install instructions:

Check both options

Add Anaconda to the system PATH environment variable
This ensures that PATH is set correctly when using Python, IPython, conda, and any other program in the Anaconda distribution. If unchecked, you must use the Anaconda Command Prompt (located in the Start Menu under "Anaconda (32-bit)").

Register Anaconda as the system Python 3.5
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.5 on the system.
Running Spyder

Click on ‘Spyder’ from Anaconda program group:
Spyder:

```python
# Python program that can be executed to report whether particular
# python packages are available on the system.

import math
import os
import sys

# Let's stop here
sys.exit(1)
return None

# assert major == 3, "Stopping here - we need Python 3."
if minor >= 5:
    print("Testing Python version-> py{}(). OK".format(major, minor))
else:
    print("Warning: You should be running Python 3.5 or newer, " +
    "you have Python {}().".format(major, minor))
```

Alvaro Sebastián Yagüe

www.sixthresearcher.com

@SixthResearcher
Running Jupyter Notebook

Click on ‘Jupyter Notebook’ from Anaconda group:
Anaconda

Jupyter Notebook:
Python for scientists

Next lesson...
Python scripts
and modules

```python
def complementary(seq):
    nt_comp = {
        'A': 'T',
        'C': 'G',
        'G': 'C',
        'T': 'A',
    }
    for nt in seq:
        compseq += nt_comp[nt]
    return compseq
```