Python for scientists

Python, Spyder, Jupyter and Anaconda

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.

http://www.southampton.ac.uk/~fangohr/blog/installation-of-python-spyder-numpy-sympy-scipy-pytest-matplotlib-via-anaconda.html

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 (currently 96220).

http://www.southampton.ac.uk/~fangohr/blog/installation-of-python-spyder-numpy-sympy-scipy-pytest-matplotlib-via-anaconda.html

Spyder

Spyder

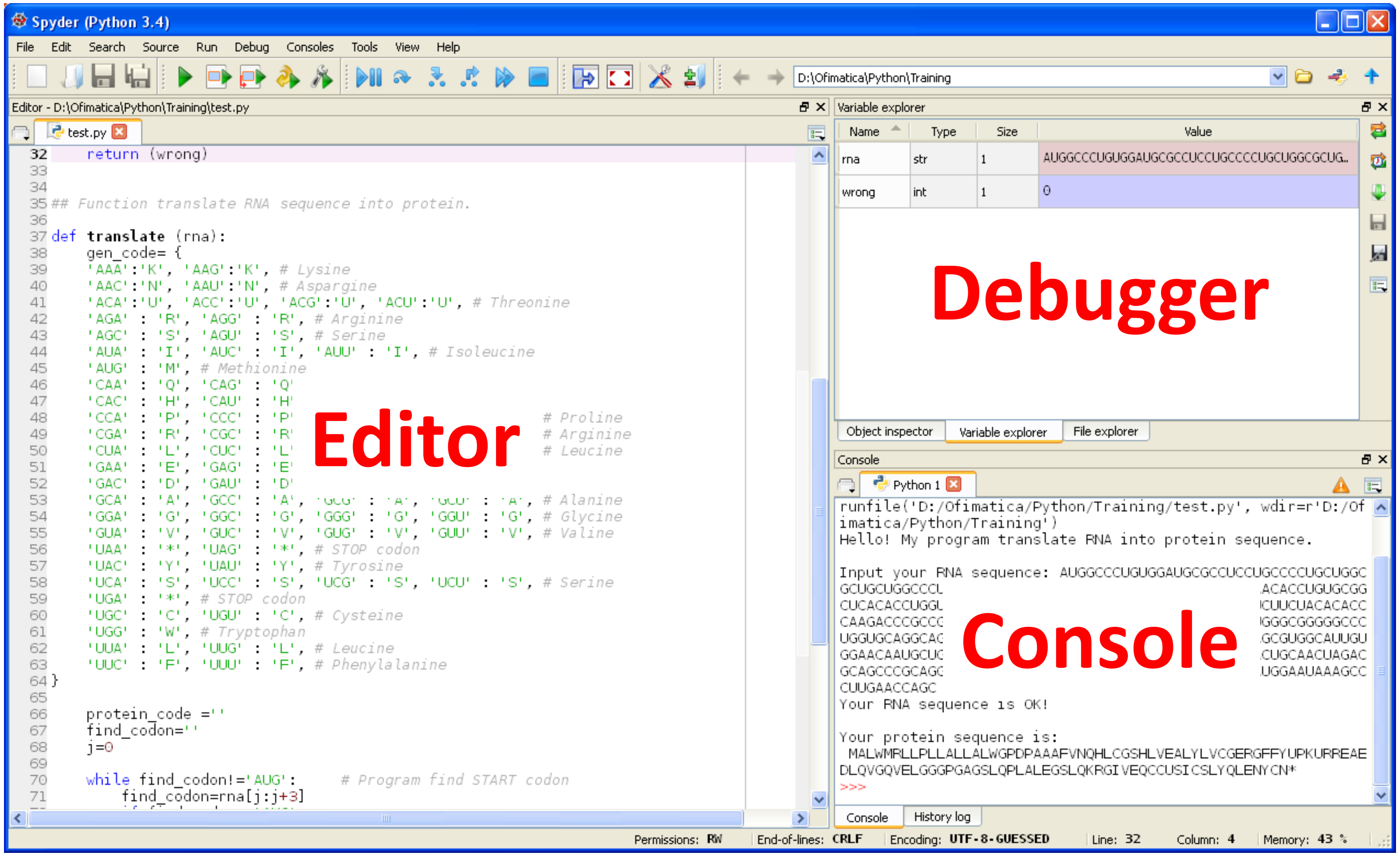
Spyder **is s a powerful IDE (interactive development environment)** for the Python language with advanced editing, interactive testing, debugging and introspection features.

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



```python
    return (wrong)


## Function translate RNA sequence into protein.

def translate (rna):
    gen_code= {
    'AAA':'K', 'AAG':'K', # Lysine
    'AAC':'N', 'AAU':'N', # Aspargine
    'ACA':'U', 'ACC':'U', 'ACG':'U', 'ACU':'U', # Threonine
    'AGA' : 'R', 'AGG' : 'R', # Arginine
    'AGC' : 'S', 'AGU' : 'S', # Serine
    'AUA' : 'I', 'AUC' : 'I', 'AUU' : 'I', # Isoleucine
    'AUG' : 'M', # Methionine
    'CAA' : 'Q', 'CAG' : 'Q'
    'CAC' : 'H', 'CAU' : 'H'
    'CCA' : 'P', 'CCC' : 'P'                                  # Proline
    'CGA' : 'R', 'CGC' : 'R'                                  # Arginine
    'CUA' : 'L', 'CUC' : 'L'                                  # Leucine
    'GAA' : 'E', 'GAG' : 'E'
    'GAC' : 'D', 'GAU' : 'D'
    'GCA' : 'A', 'GCC' : 'A', 'GCG' : 'A', 'GCU' : 'A', # Alanine
    'GGA' : 'G', 'GGC' : 'G', 'GGG' : 'G', 'GGU' : 'G', # Glycine
    'GUA' : 'V', 'GUC' : 'V', 'GUG' : 'V', 'GUU' : 'V', # Valine
    'UAA' : '*', 'UAG' : '*', # STOP codon
    'UAC' : 'Y', 'UAU' : 'Y', # Tyrosine
    'UCA' : 'S', 'UCC' : 'S', 'UCG' : 'S', 'UCU' : 'S', # Serine
    'UGA' : '*', # STOP codon
    'UGC' : 'C', 'UGU' : 'C', # Cysteine
    'UGG' : 'W', # Tryptophan
    'UUA' : 'L', 'UUG' : 'L', # Leucine
    'UUC' : 'F', 'UUU' : 'F', # Phenylalanine
}

    protein_code =''
    find_codon=''
    j=0

    while find_codon!='AUG':     # Program find START codon
        find_codon=rna[j:j+3]
```

Name	Type	Size	Value
rna	str	1	AUGGCCCUGUGGAUGCGCCUCCUGCCCCUGCUGGCGCUG..
wrong	int	1	0

```
runfile('D:/Ofimatica/Python/Training/test.py', wdir=r'D:/Of
imatica/Python/Training')
Hello! My program translate RNA into protein sequence.

Input your RNA sequence: AUGGCCCUGUGGAUGCGCCUCCUGCCCCUGCUGGC
GCUGCUGGCCCL                                      ACACCUGUGCGG
CUCACACCUGGL                                      CUUCUACACACC
CAAGACCCGCCC                                      GGGCGGGGGCCC
UGGUGCAGGCAC                                      GCGUGGCAUUGU
GGAACAAUGCUC                                      CUGCAACUAGAC
GCAGCCCGCAGC                                      UGGAAUAAAGCC
CUUGAACCAGC
Your RNA sequence is OK!

Your protein sequence is:
 MALWMRLLPLLALLALWGPDPAAAFVNQHLCGSHLVEALYLVCGERGFFYUPKURREAE
DLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQCCUSICSLYQLENYCN*
>>>
```

Permissions: RW   End-of-lines: CRLF   Encoding: UTF-8-GUESSED   Line: 32   Column: 4   Memory: 43 %

Alvaro Sebastián Yagüe

www.sixthresearcher.com

@SixthResearcher

Jupyter Notebook

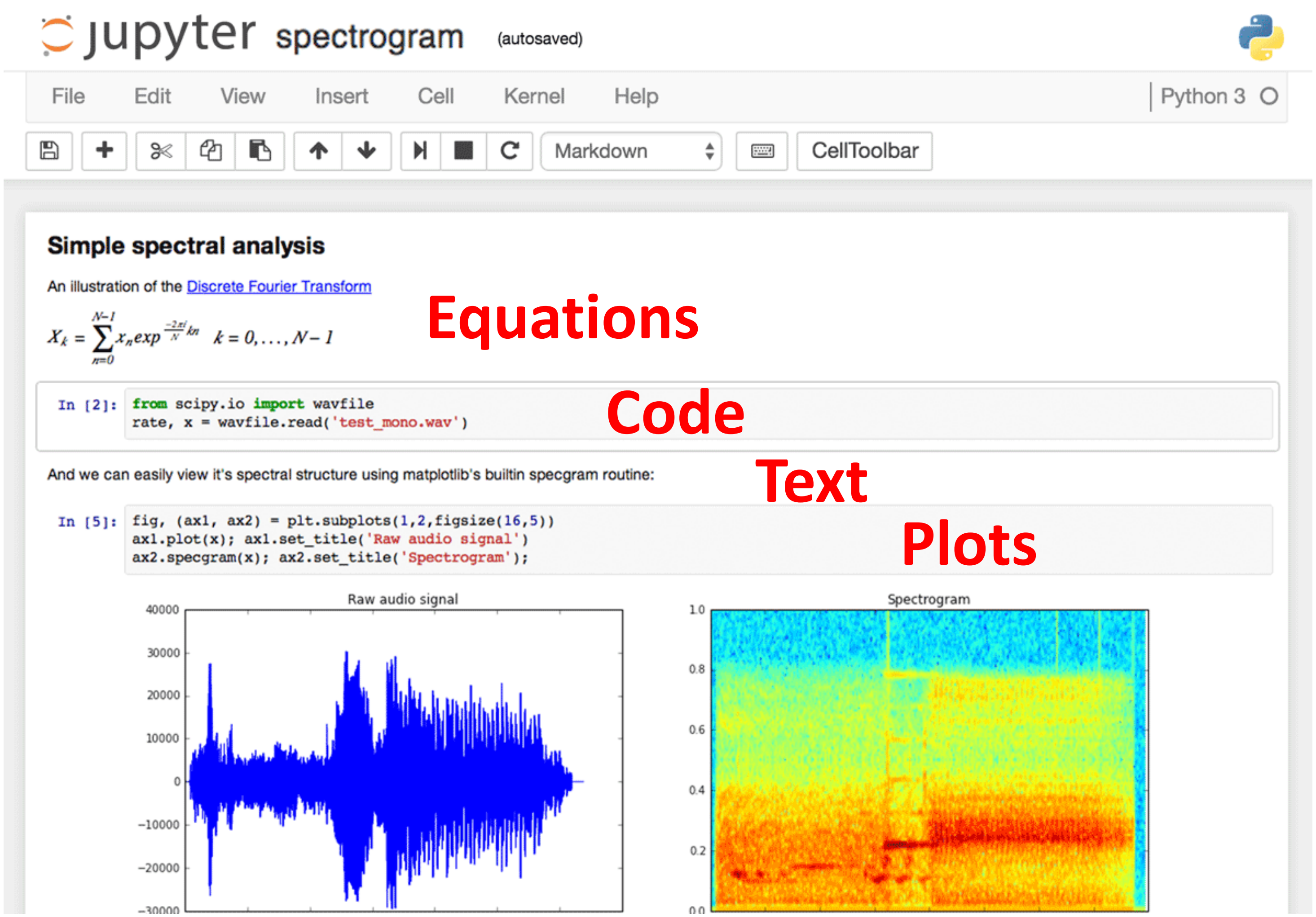
Jupyter Notebook (former IPython)

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

$$X_k = \sum_{n=0}^{N-1} x_n exp^{\frac{-2\pi i}{N} kn} \quad k = 0, \ldots, N-1$$

Equations

```
In [2]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

Code

And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

Text

```
In [5]: 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');
```

Plots

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.

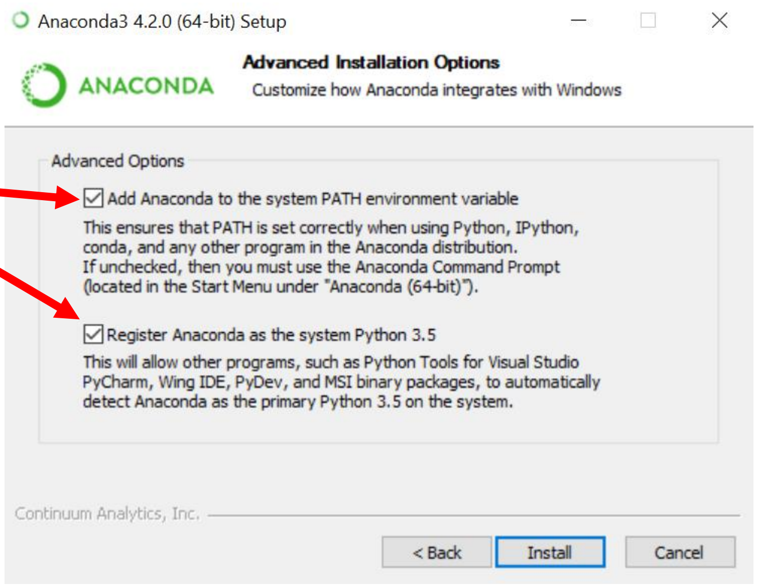
Downloading and installing Anaconda

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


Anaconda3 4.2.0 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced 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, then you must use the Anaconda Command Prompt (located in the Start Menu under "Anaconda (64-bit)").

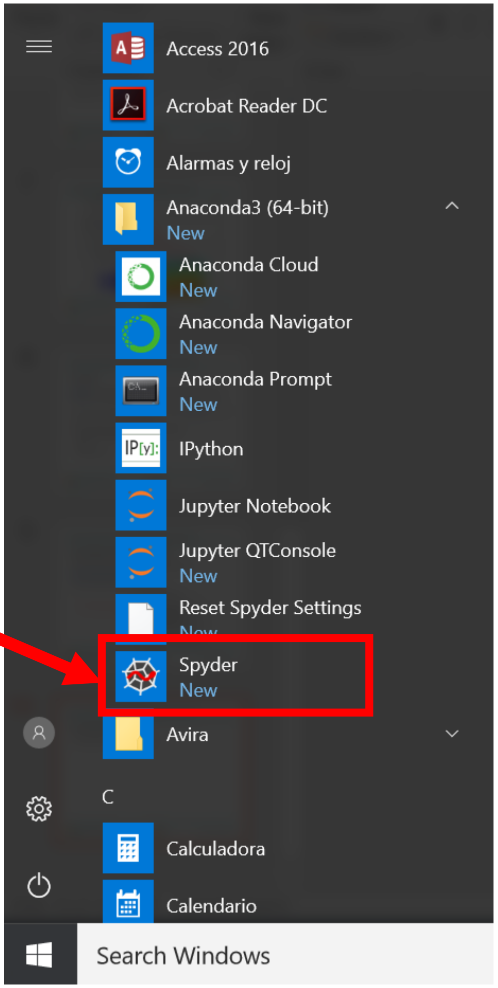
[✓] Register Anaconda as the system Python 3.5

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.

Anaconda

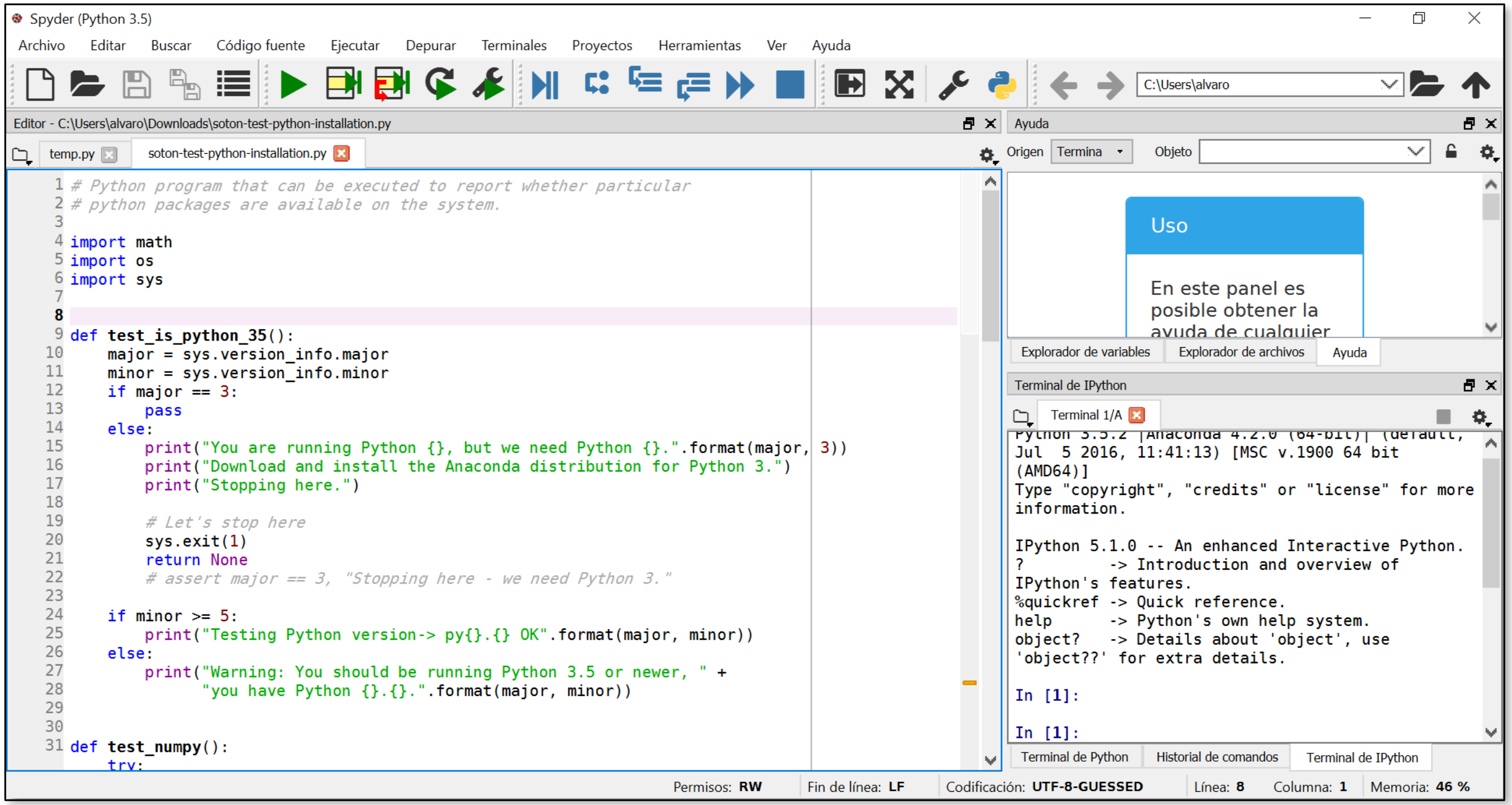
Running Spyder

Click on 'Spyder' from Anaconda program group:



Spyder

Spyder:


Spyder (Python 3.5)
Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda
C:\Users\alvaro
Editor - C:\Users\alvaro\Downloads\soton-test-python-installation.py
temp.py
soton-test-python-installation.py
Ayuda
Origen Termina Objeto
Uso
En este panel es posible obtener la ayuda de cualquier
Explorador de variables Explorador de archivos Ayuda
Terminal de IPython
Terminal 1/A
Terminal de Python Historial de comandos Terminal de IPython
Permisos: RW Fin de línea: LF Codificación: UTF-8-GUESSED Línea: 8 Columna: 1 Memoria: 46 %

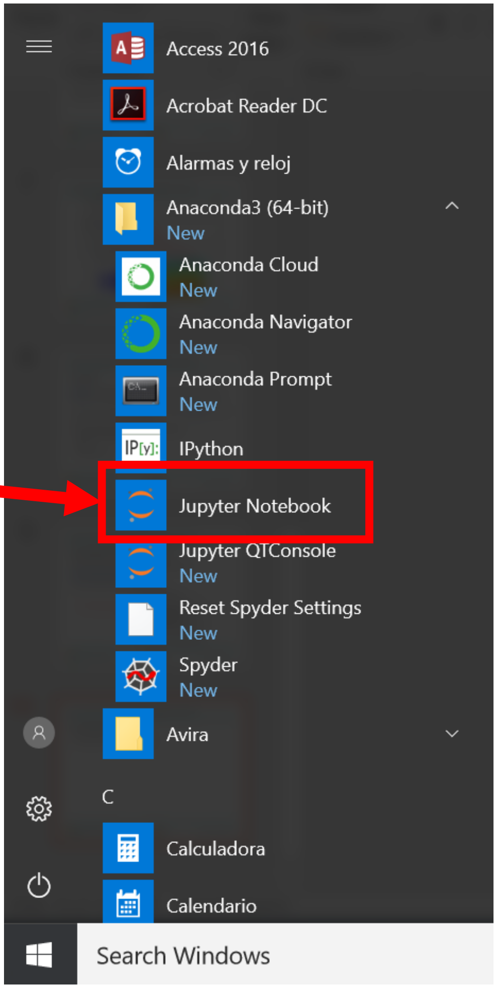
```python
# Python program that can be executed to report whether particular
# python packages are available on the system.

import math
import os
import sys


def test_is_python_35():
    major = sys.version_info.major
    minor = sys.version_info.minor
    if major == 3:
        pass
    else:
        print("You are running Python {}, but we need Python {}.".format(major, 3))
        print("Download and install the Anaconda distribution for Python 3.")
        print("Stopping here.")

        # 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))


def test_numpy():
    try:
```

```
Python 3.5.2 |Anaconda 4.2.0 (64-bit)| (default,
Jul  5 2016, 11:41:13) [MSC v.1900 64 bit
(AMD64)]
Type "copyright", "credits" or "license" for more
information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of
IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use
'object??' for extra details.

In [1]:

In [1]:
```

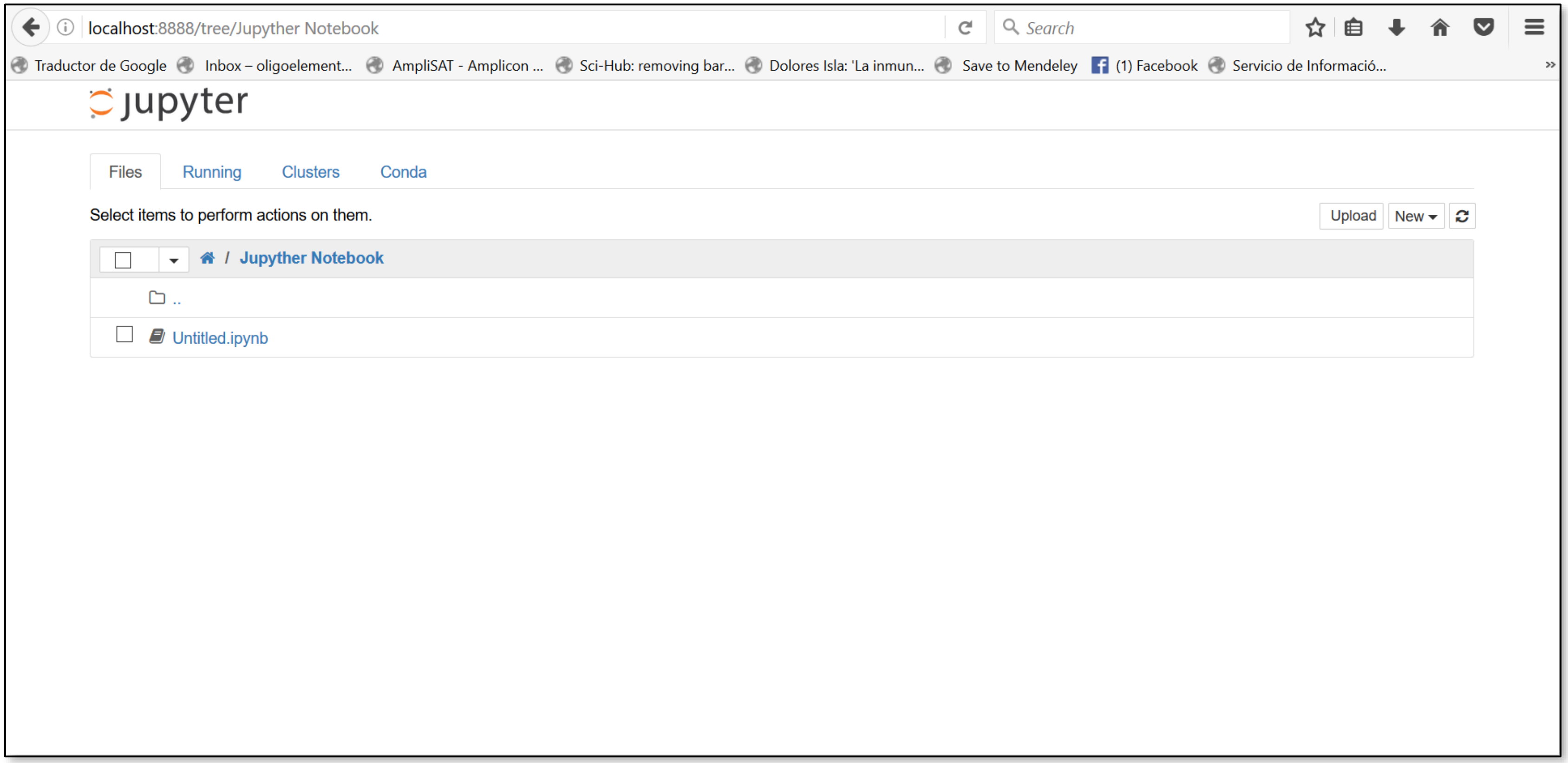
Running Jupyter Notebook

Click on 'Jupyter Notebook' from Anaconda group:

Jupyter Notebook



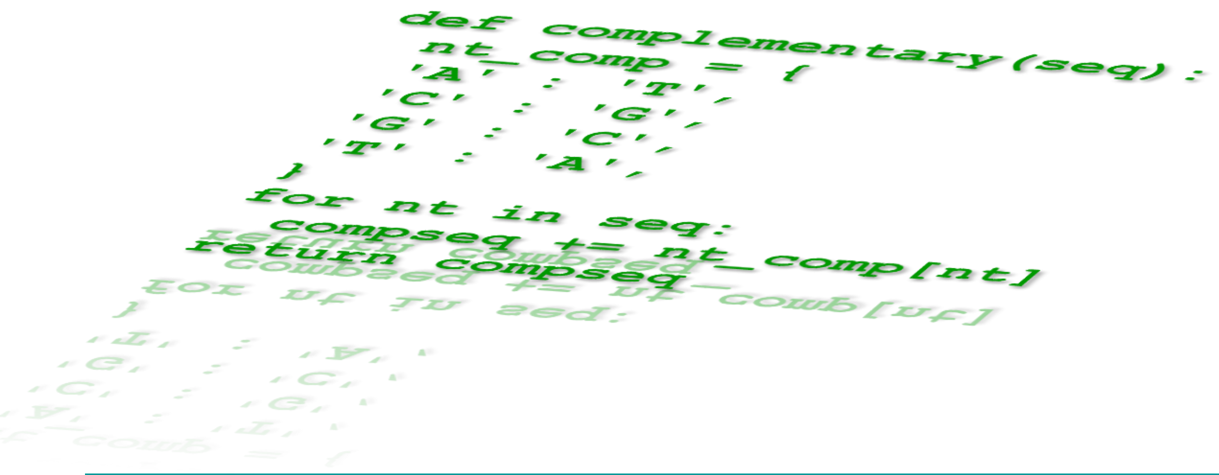
Jupyter Notebook:



Select items to perform actions on them.

Python for scientists

Next lesson...
Python scripts and modules



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

Alvaro Sebastián Yagüe

www.sixthresearcher.com

@SixthResearcher