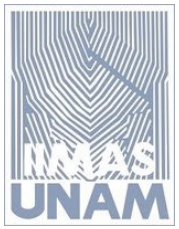
Visualizing scientific data using Python

A practical approach



Dr. Augusto César Poot Hernández.
Postdoctoral fellow.
IIMAS, UNAM
augusto.poot@iimas.unam.mx
dragopoot@gmail.com

III Summer School in Bioinformatics.
June 10-24, 2018.
Santiago de Querétaro, Querétaro.

First, an analogy...

Mono Lake, July 1993.
Ken Rockwell.
Taken with a broken camera!!

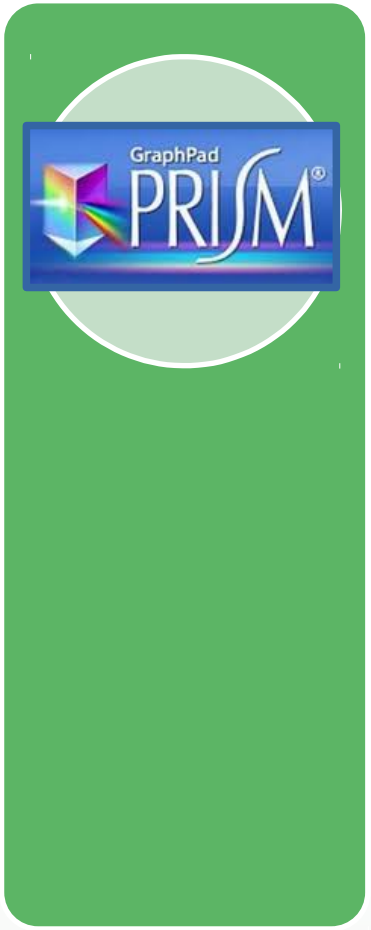
What is the best tool?

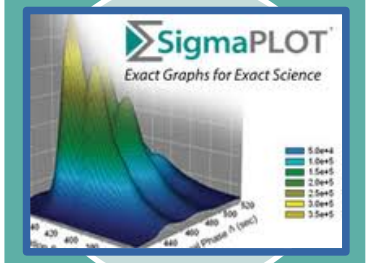
~ $ 1000 USD

~ $ 400 USD

~ $ 5000 USD

Same occurs with science.





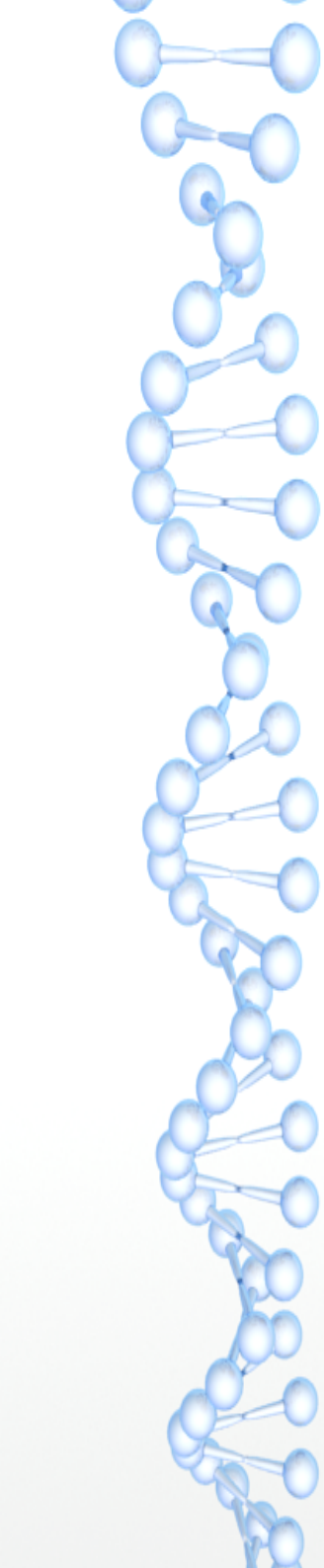
Qti Plot

Take home.

- The best tool is the one that meets your needs.
- For this reason, the choice of the best tool is (hopefully) a personal decision.
- The mastery to use that tool comes with (a lot) of practice and perseverance.

My objectives.

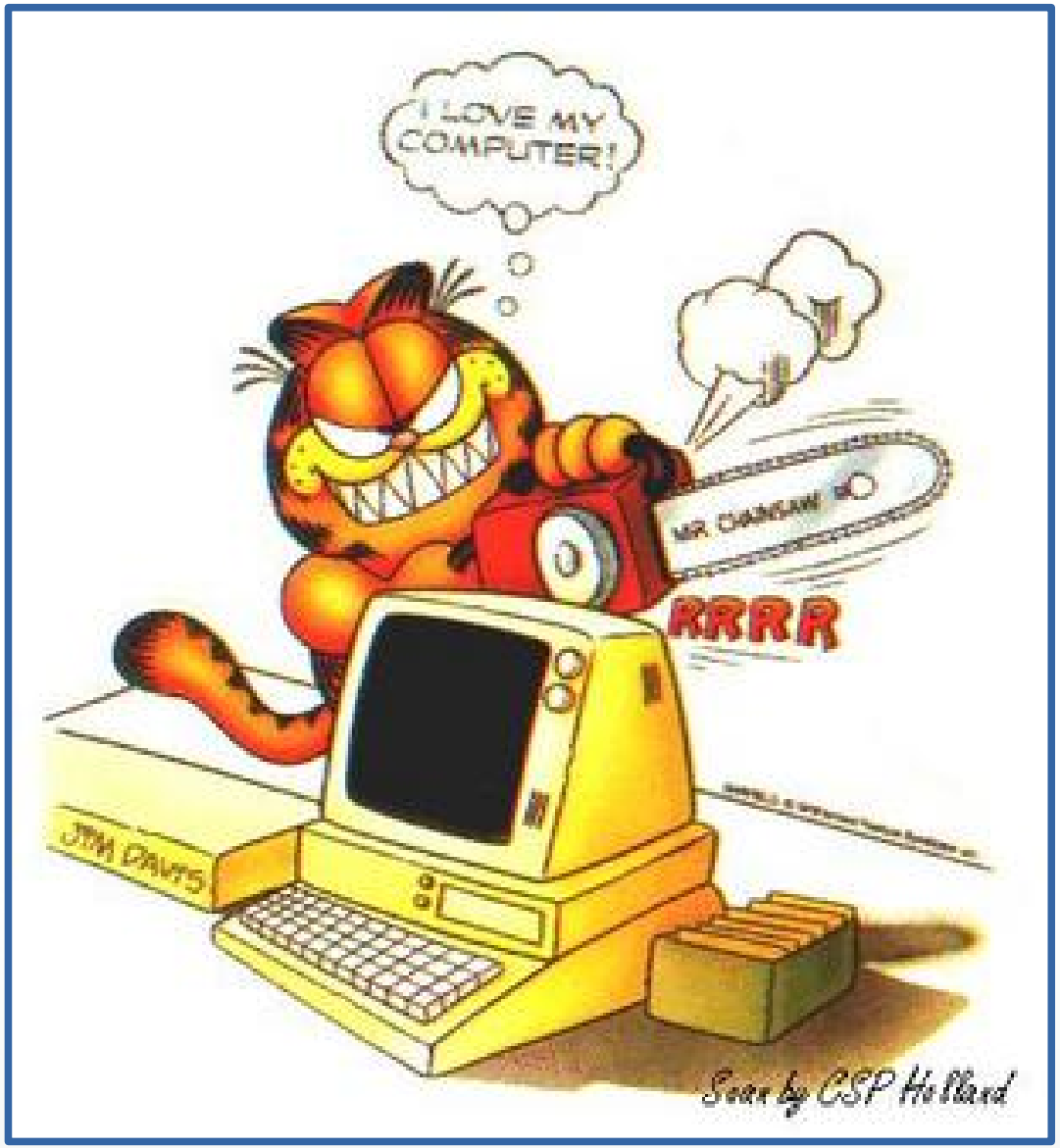
- Create a primer for the use of Python for scientific analysis.
- Convince you that python is a fun and easy to use alternative for this purpose.



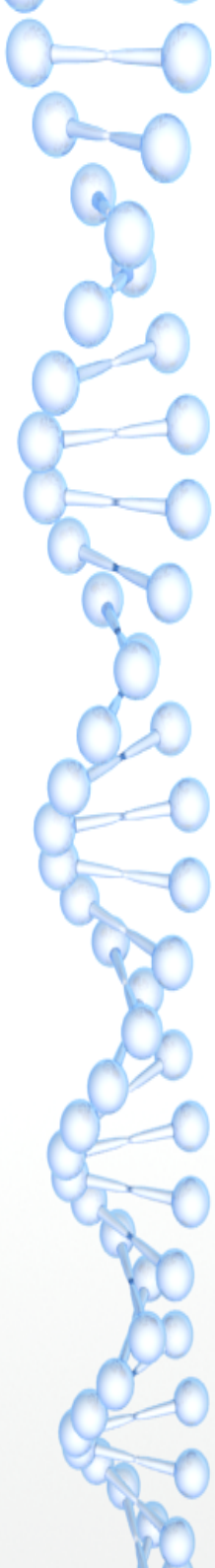
Content.

1) Python packages for scientific analysis.
2) iPython: Enhanced python interpreter.
3) Scientific python stack:
   - Numpy
   - Matplotlib
   - Scipy
   - Pandas

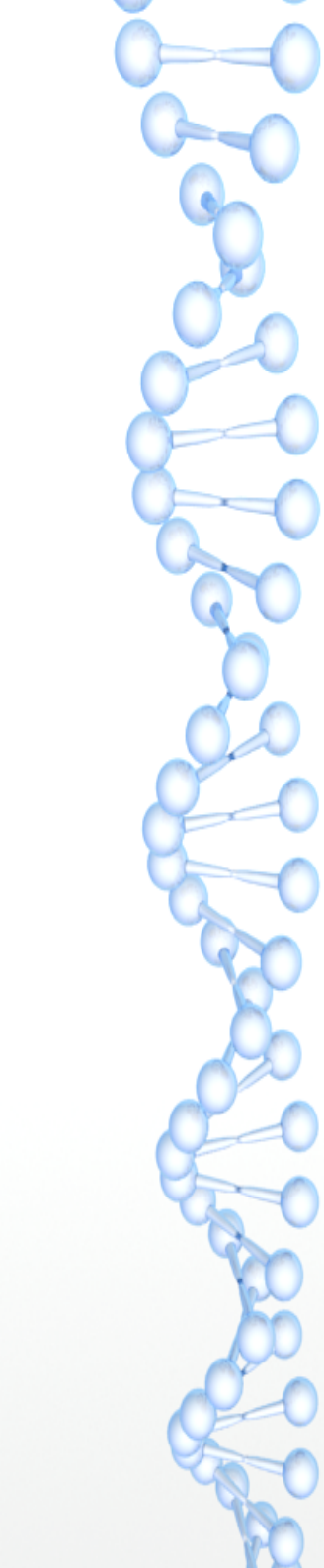
Be patient...



- Interpreted programming language
- Multi-purpose, multi-paradigm and multi-platform programming language.
- Emphasis on clean syntax.
- Excellent to learn and to prototyping.
- It is free software.
- A wide variety of (also free) modules or libraries.



It is said that Python is a programming language with Batteries Included, because the Standard Library contains enough functions for almost any application.



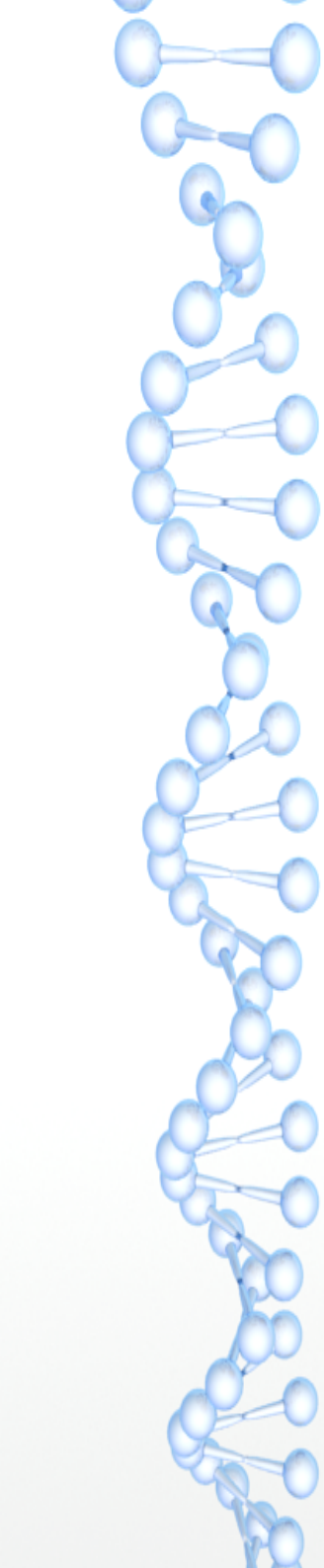
However you don’t need to re-invent the wheel. Specialized modules for certain applications (science) make work much easier and speed up code production and execution.

What is a Python library or module?

It is a file(s) that contains Python code (definitions and statements) related to certain task.

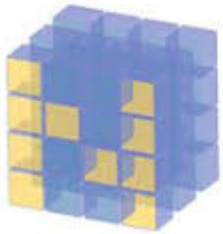
The definitions in the module can be used by other python programs.

```
In [1]: import time

In [2]: time.strftime('%d / %m / %Y at %H hrs')
Out[2]: '04 / 06 / 2018 at 10 hrs'

In [3]: import math

In [4]: math.pi
Out[4]: 3.141592653589793

In [5]:
```



There are A LOT scientific related modules for Python.

Fantastic four



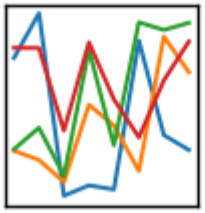
NumPy

SciPy

matplotlib

pandas

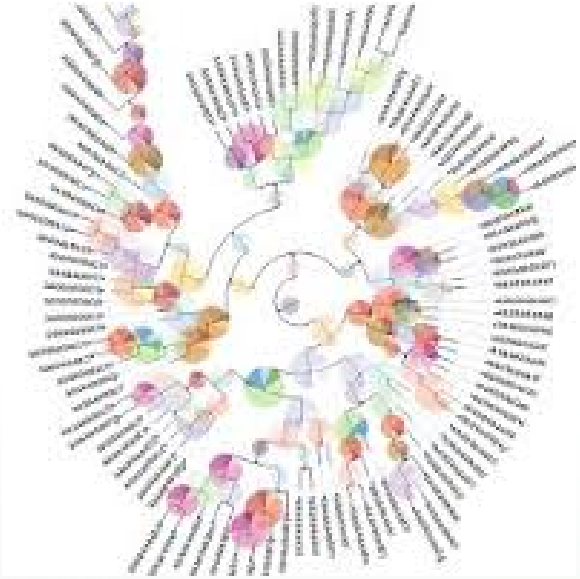
$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



The X-men

biopython

scikit-bio



ETE toolkit

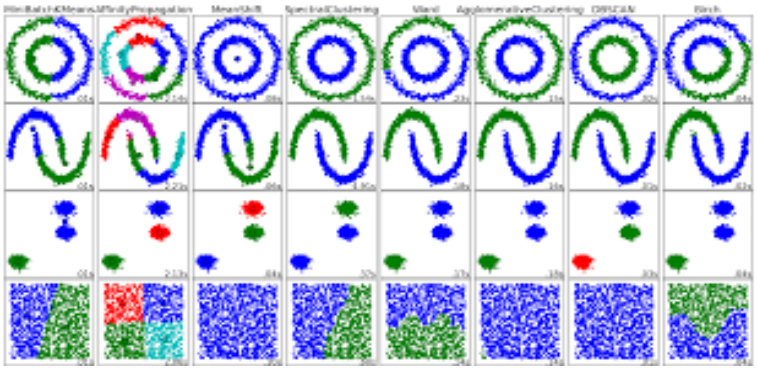
DendroPy

Epipy

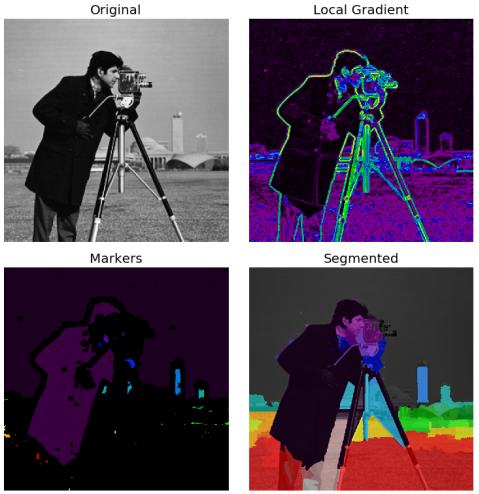
Python tools for epidemiology

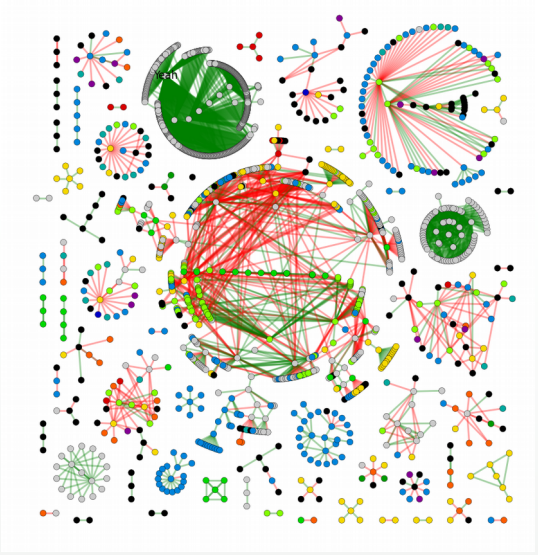
The Avengers

scikits learn
machine learning in Python



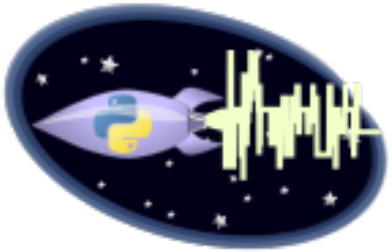
scikit-image
image processing in python





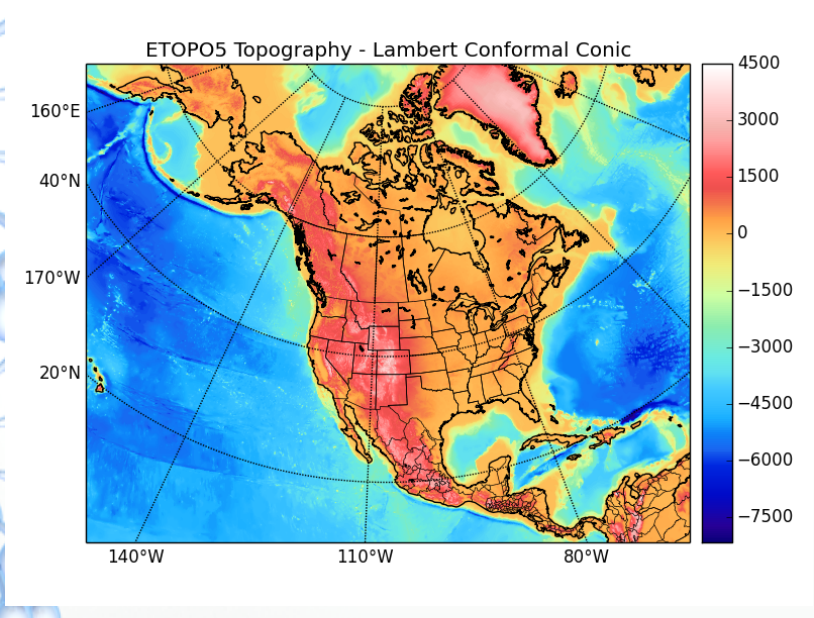
NetworkX

StatsModels
Statistics in Python

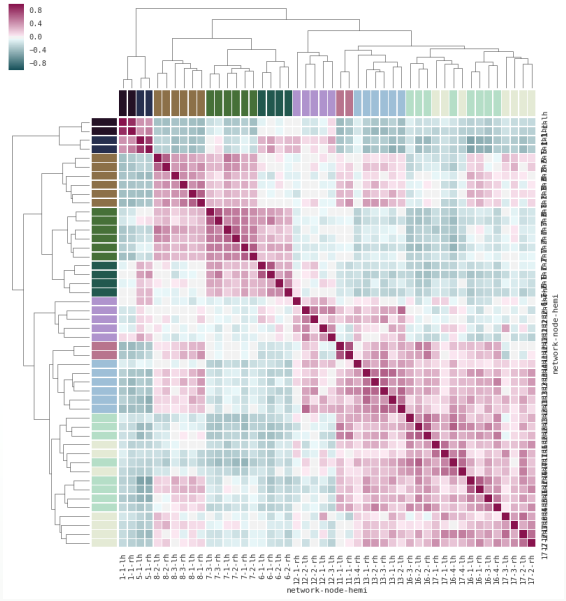


PyMC

Basemap



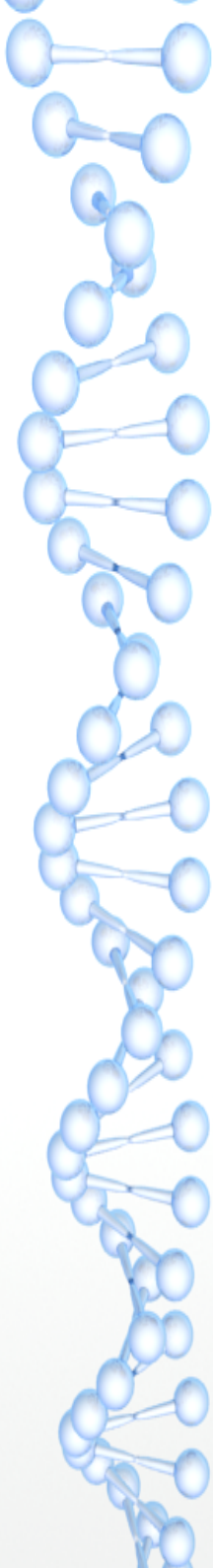
Seaborn



On the edge of machine learning...

Artificial Neural Networks

API for python



And a lot more...

How are we going to work?

- We will follow script-like files (.py) that contain the code to test in class.
  - !!! These files are not written to be executed as a script.
- We will execute (write) the lines of code interactively on iPython interpreter.
- We will explain the statements and the results.

Python 2 Vs Python 3

(Some differences)

```
In [1]: print "Hello World"
Hello World
```

```
In [1]: print("Hello World")
Hello World
```

print

```
In [2]: 3/2
Out[2]: 1

In [3]: 3./2
Out[3]: 1.5
```

```
In [2]: 3/2
Out[2]: 1.5

In [3]: 3//2
Out[3]: 1
```

division

```
In [4]: range(0,10)
Out[4]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

In [5]: xrange(0,10)
Out[5]: xrange(10)
```

```
In [4]: range(0,10)
Out[4]: range(0, 10)

In [5]: list(range(0,10))
Out[5]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

range

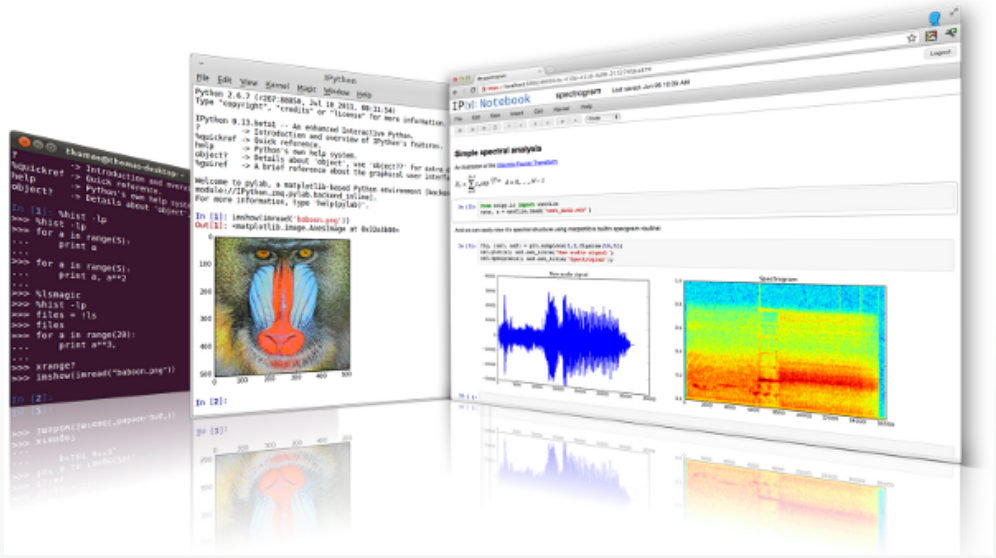
```
In [6]: dicta = {'a': 'first', 'b': 'second'}

In [7]: dicta.keys()
Out[7]: ['a', 'b']
```

```
In [6]: dicta = {'a': 'first', 'b': 'second'}

In [7]: dicta.keys()
Out[7]: dict_keys(['a', 'b'])

In [8]: list(dicta.keys())
Out[8]: ['a', 'b']
```

dictionaries

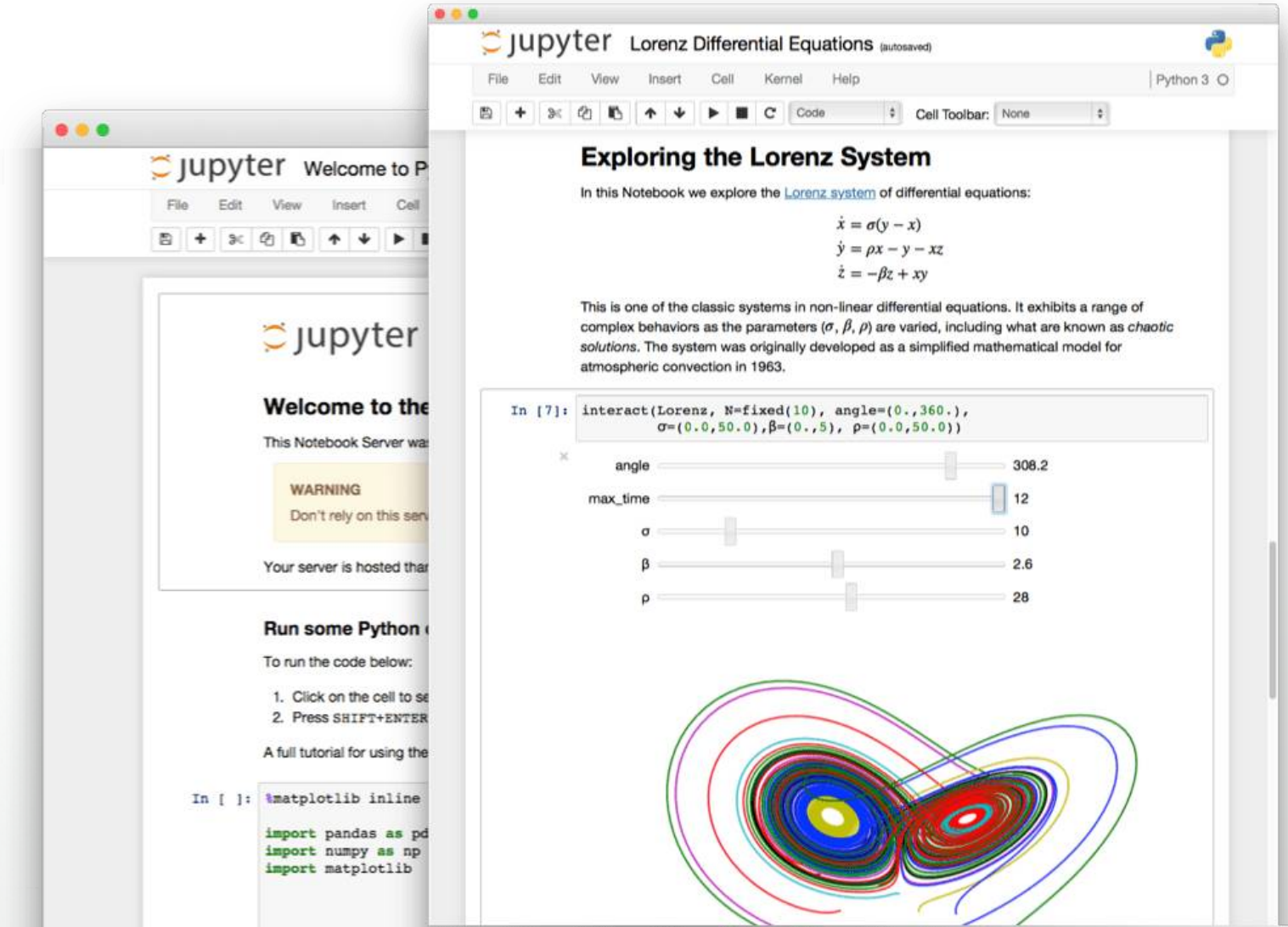
IP[y]: IPython
Interactive Computing

- A powerful interactive shell for Python.
- Interactive data visualization.
- Easy to use, high performance tools for parallel computing.

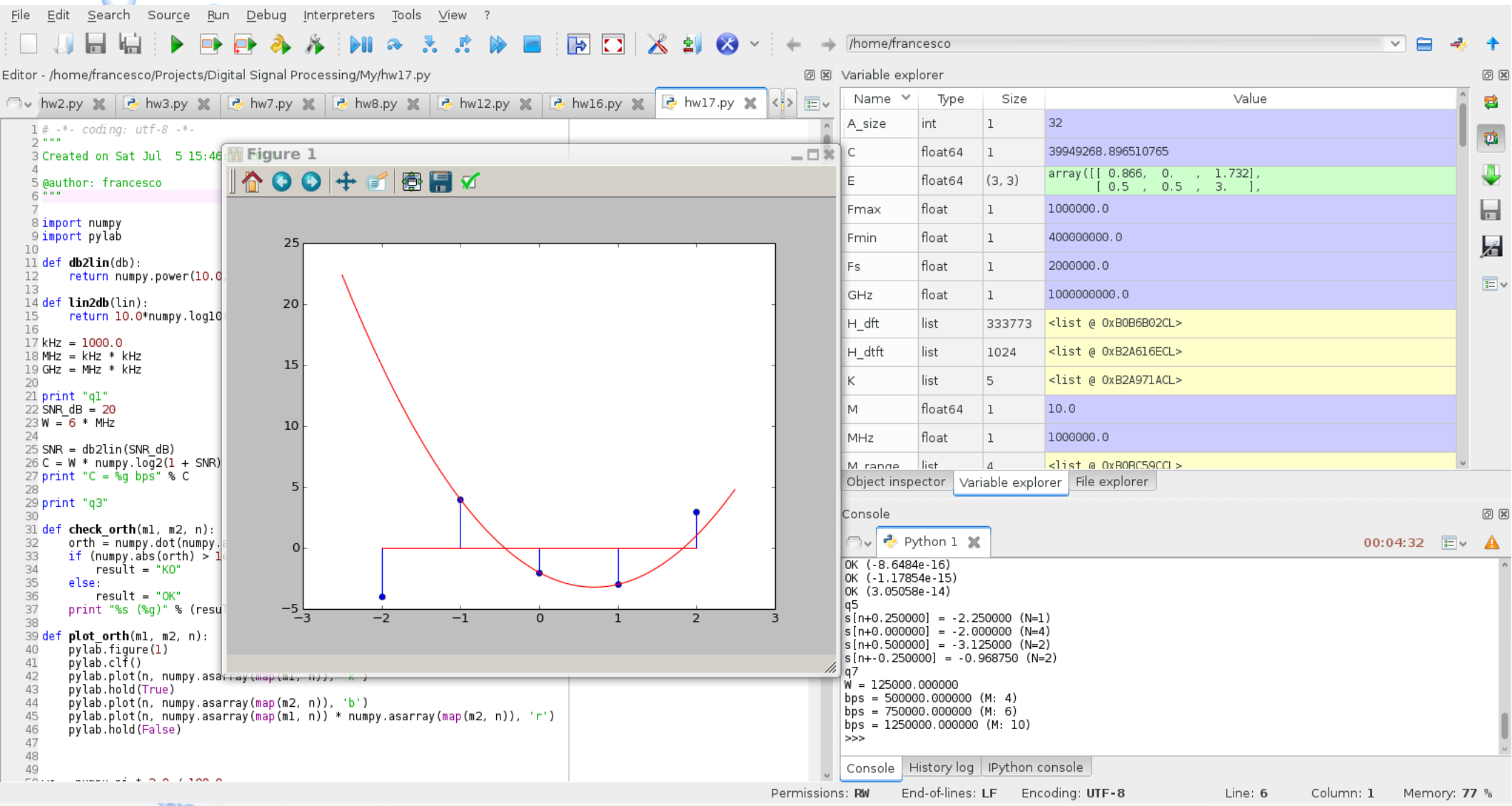


Jupyter notebooks

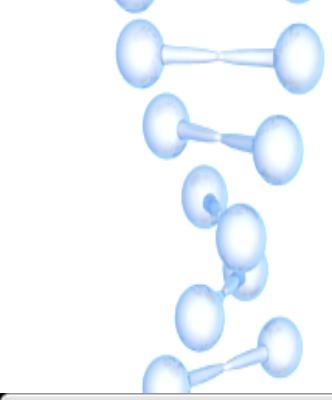
Open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.



Scientific PYthon Development EnviRonment.

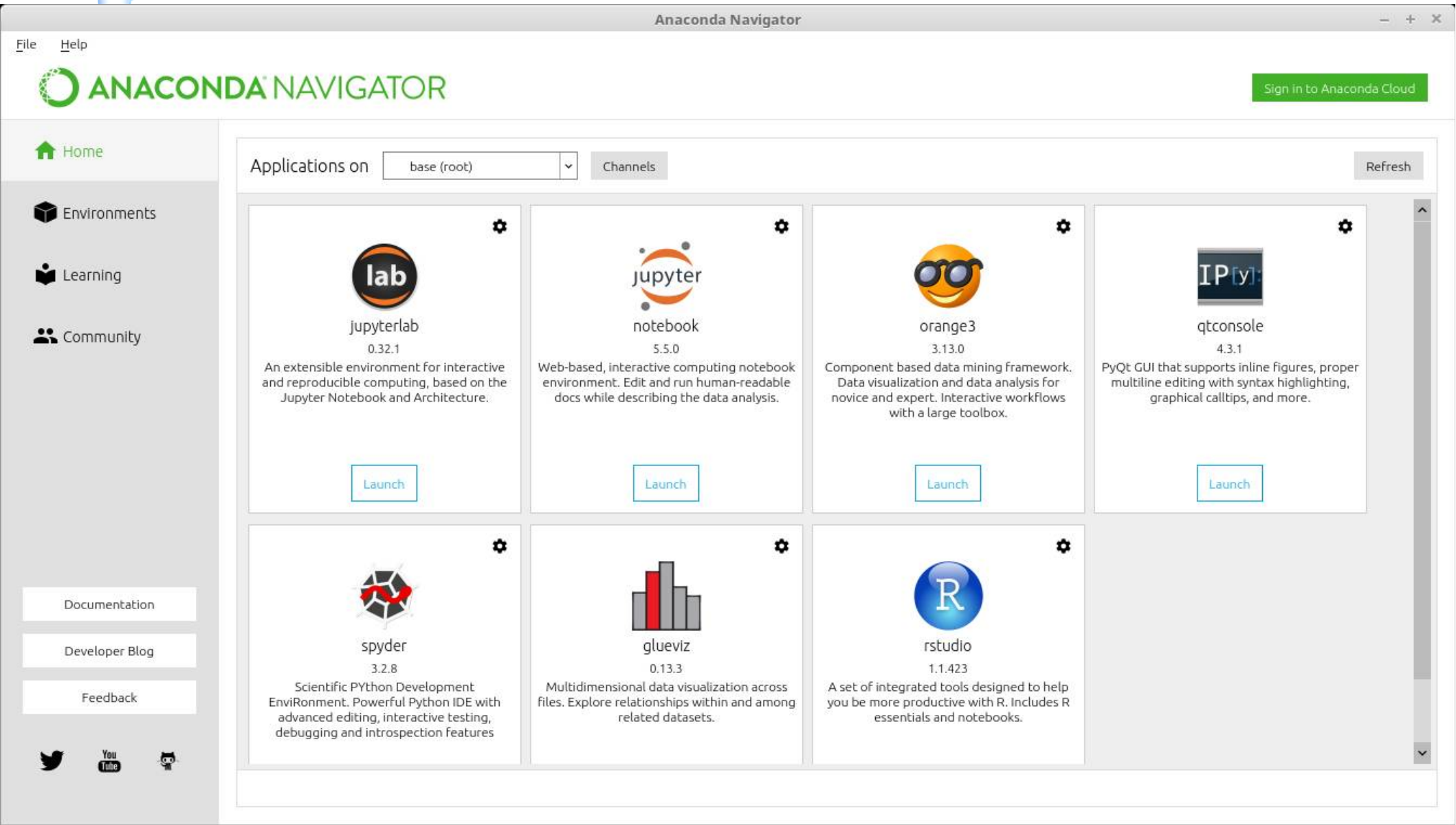


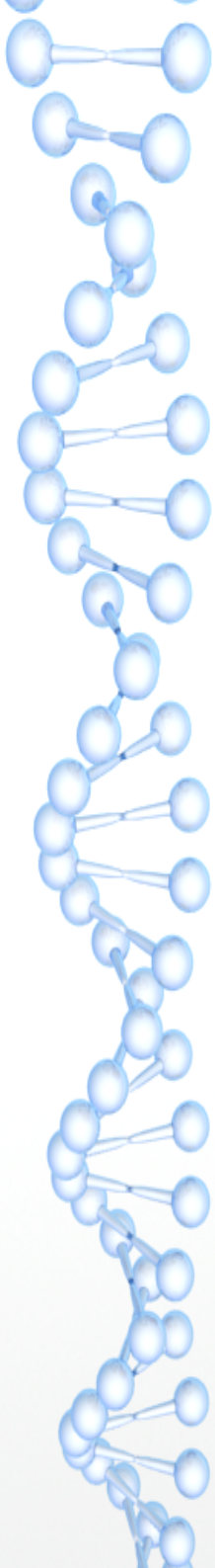
https://balau82.wordpress.com/2014/07/20/digital-signal-processing-course-on-coursera-ended/



ANACONDA
Powered by Continuum Analytics

Ptython distribution that includes hundreds of packages related to science.

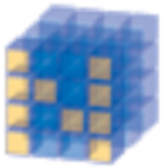




The Fantastic Four

The scipy scientific environment.

SciPy (pronounced “Sigh Pie”) is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:



NumPy

Base N-dimensional array package

SciPy library

Fundamental library for scientific computing

Matplotlib

Comprehensive 2D Plotting

IP[y]: IPython

IPython

Enhanced Interactive Console

Sympy

Symbolic mathematics

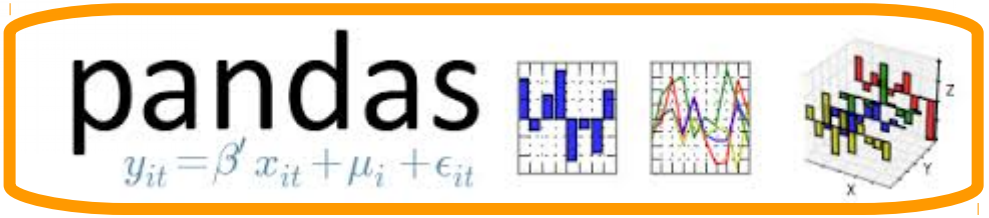
pandas

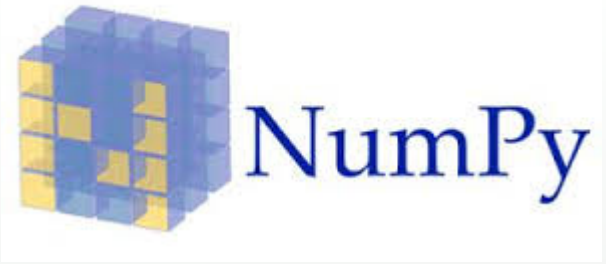
Data structures & analysis

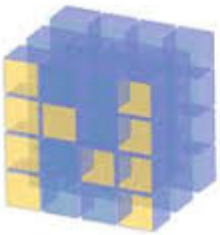
https://www.scipy.org/

Numpy stack

And more!!!



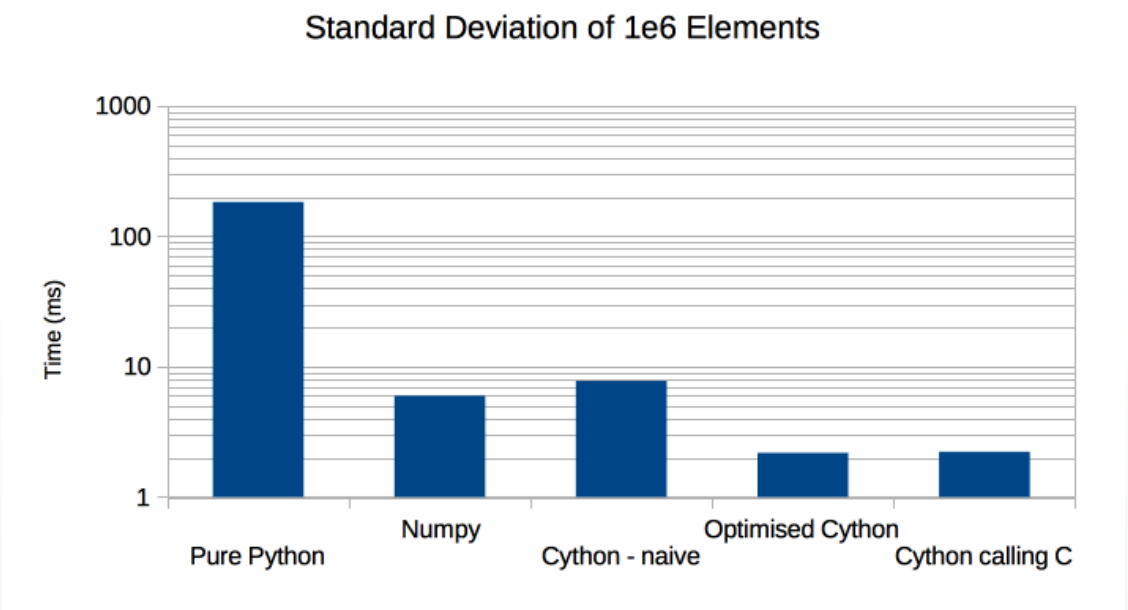




NumPy

N-dimensional numeric arrays (*ndarrays*).

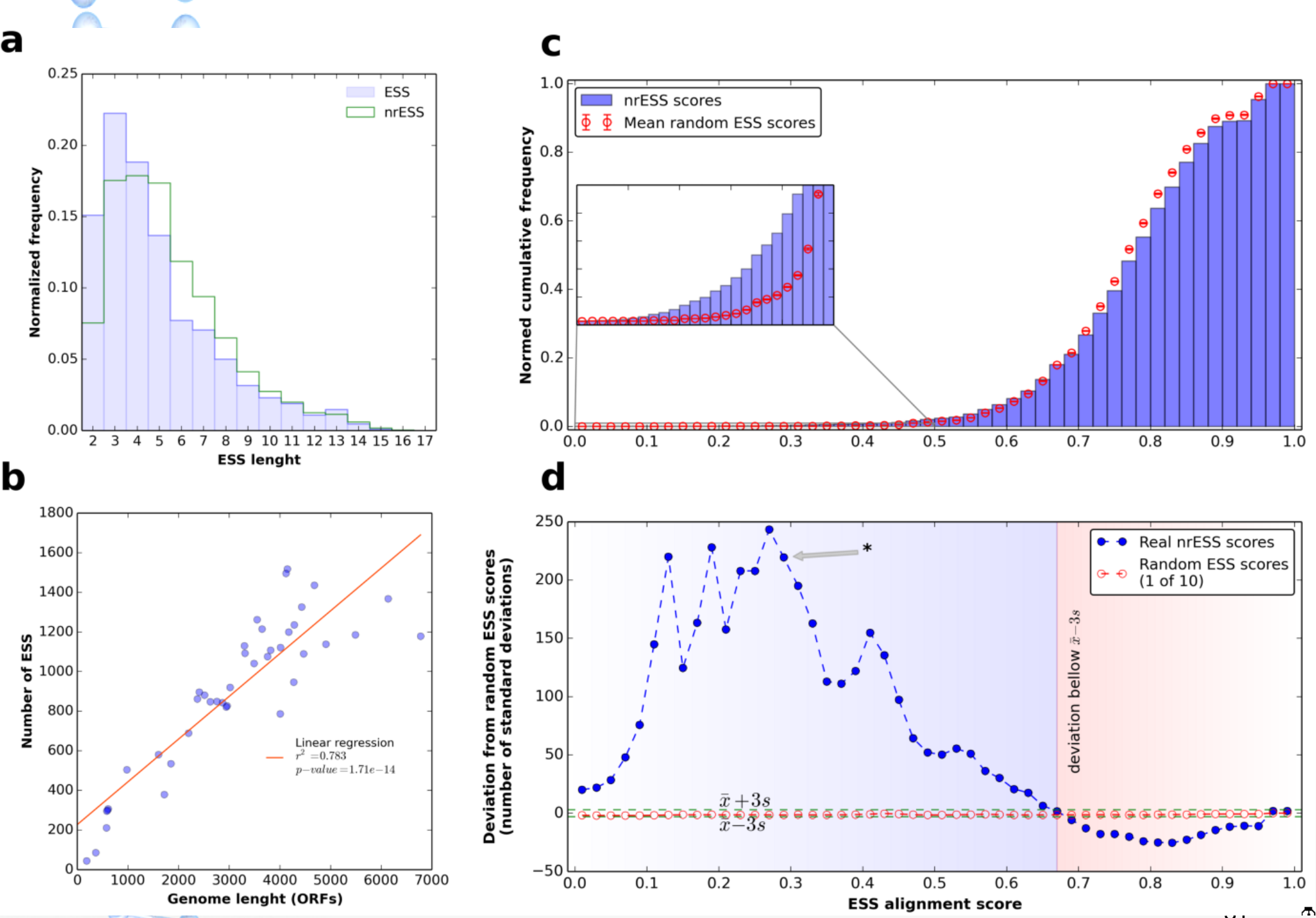
- Implemented in c/c++ , fortran.
- Functions related to *ndarrays*.
- Linear algebra, FFT, random number capabilities.



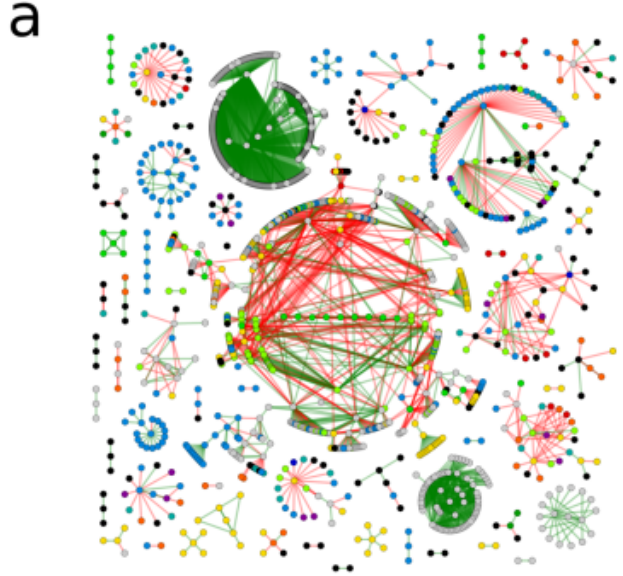
http://notes-on-cython.readthedocs.io/en/latest/std_dev.html

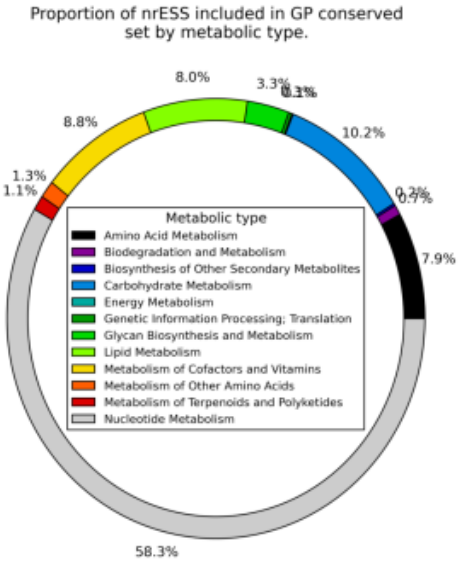
- Matplotlib is a Python 2D **plotting library** which produces **publication quality figures** in a variety of hardcopy formats and **interactive environments** across platforms.

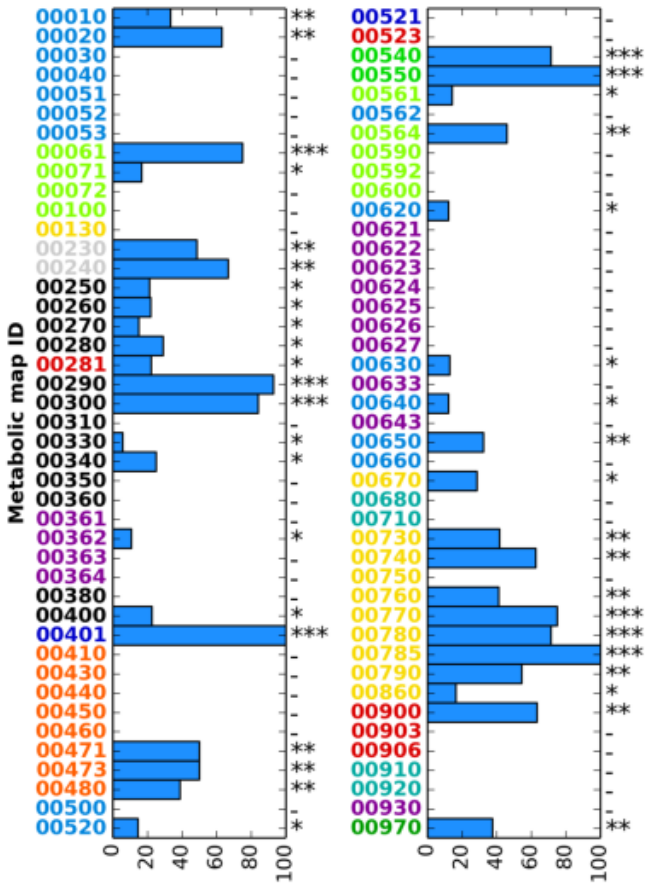
The matplotlib website at:
https://matplotlib.org/index.html

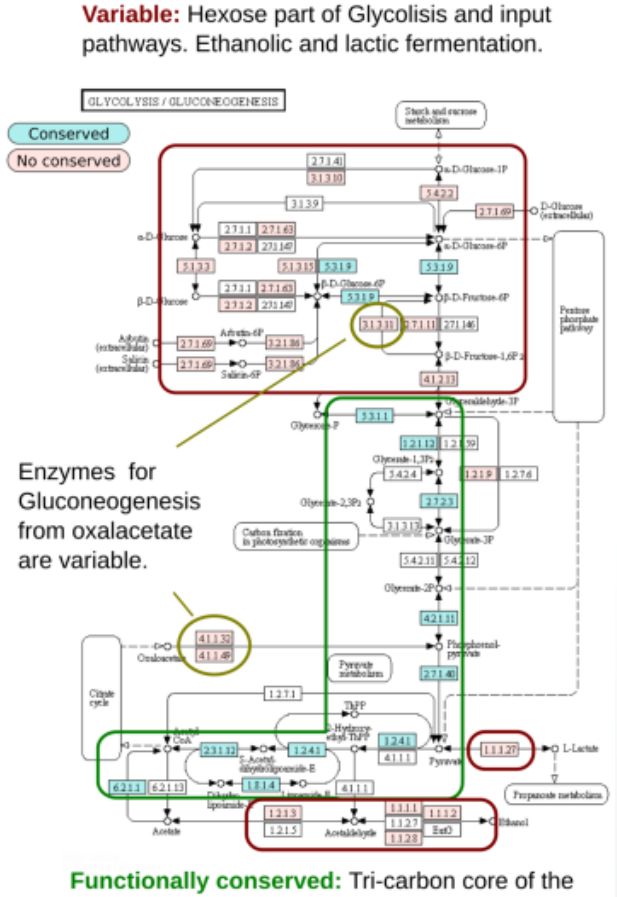


Poot-Hernandez, *et al*. 2015. BMC genomics. 16.

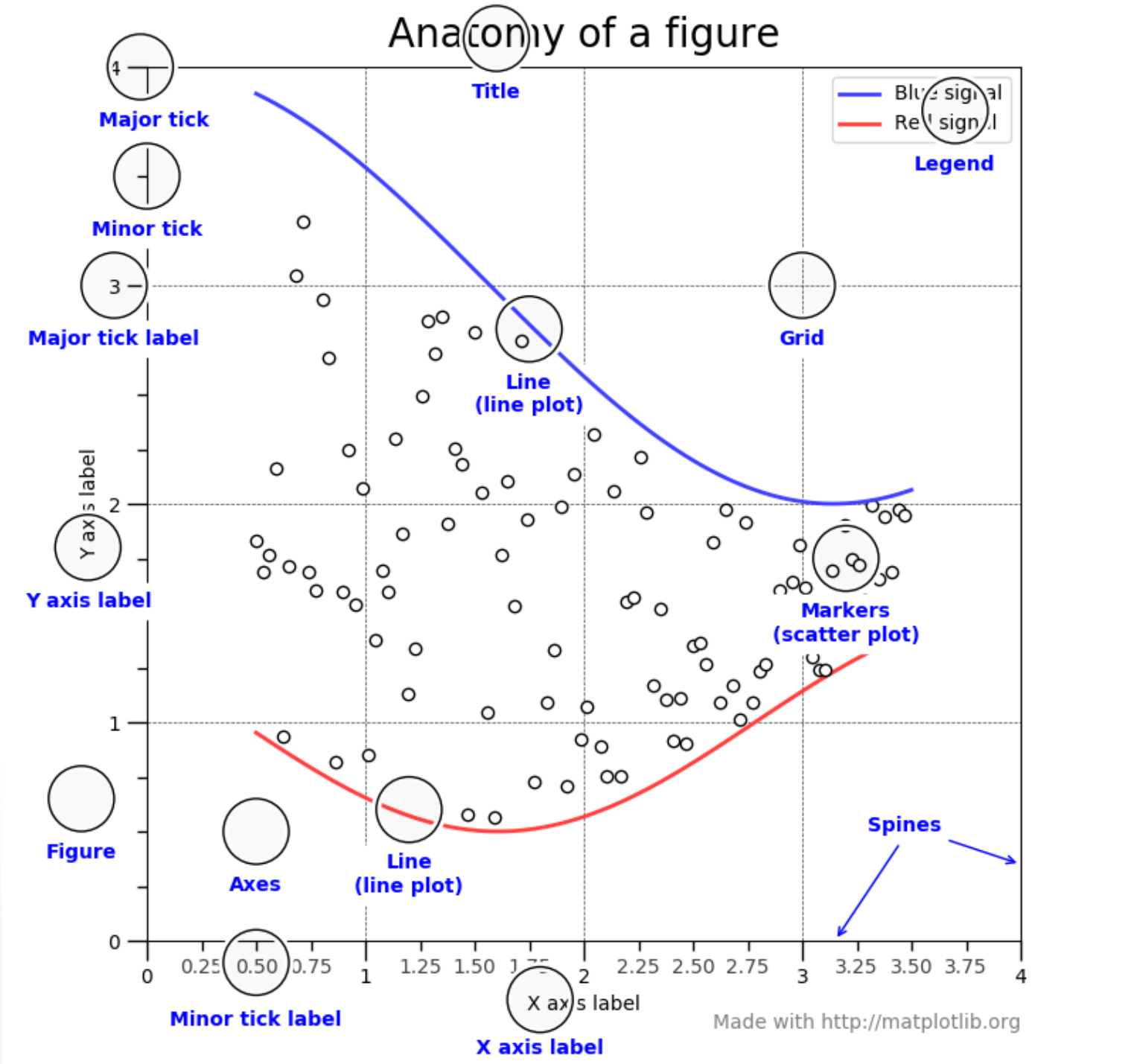








Poot-Hernandez, *et al*. 2015. BMC genomics. 16.


Anatomy of a figure
Title
Major tick
Minor tick
Major tick label
Legend
Grid
Line
(line plot)
Markers
(scatter plot)
Y axis label
Figure
Axes
Line
(line plot)
Spines
Minor tick label
X axis label
Made with http://matplotlib.org

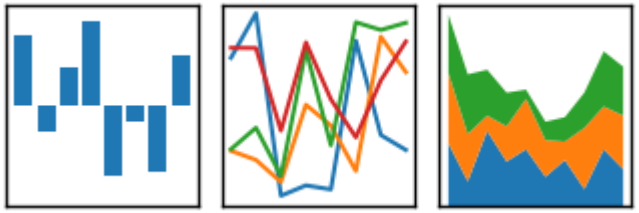
Additional routines needed in scientific work

- **constants:** physical constants and conversion factors
- **cluster:** hierarchical clustering, vector quantization, K-means
- **fftpack:** Discrete Fourier Transform algorithms
- **integrate:** numerical integration routines
- **interpolate:** interpolation tools
- **linalg:** linear algebra routines
- **misc:** miscellaneous utilities (e.g. image reading/writing)
- **ndimage:** various functions for multi-dimensional image processing
- **optimize:** optimization algorithms including linear programming
- **signal:** signal processing tools
- **sparse:** sparse matrix and related algorithms
- **spatial:** KD-trees, nearest neighbors, distance functions
- **special:** special functions
- **stats:** statistical functions
- **weave:** tool for writing C/C++ code as Python multiline strings

en.wikipedia.org

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Data structures and data analysis tools.

Data structures at a glance		
Dimensions	**Name**	**Description**
1	Series	1D labeled homogeneously-typed array
2	DataFrame	General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed columns
3	Panel	General 3D labeled, also size-mutable array

Pandas documentation at:
http://pandas.pydata.org

Pandas DataFrame

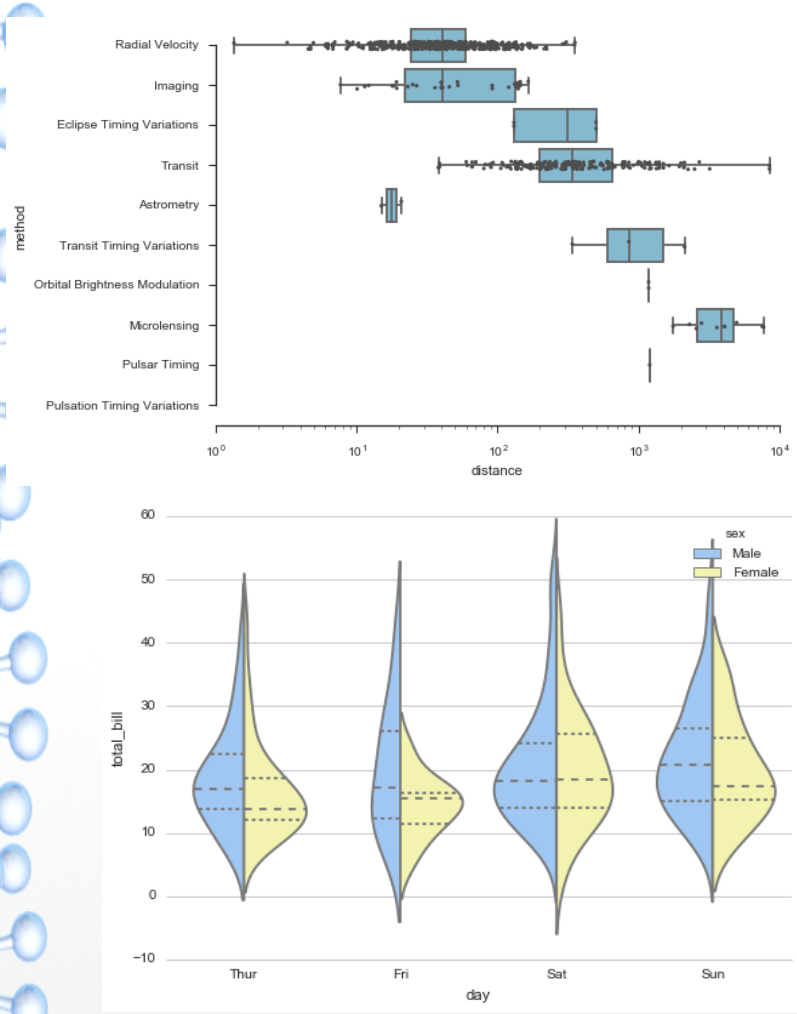
	A	B	C	D
2013-01-01	0.469112	-0.282863	-1.509059	-1.135632
2013-01-02	1.212112	-0.173215	0.119209	-1.044236
2013-01-03	-0.861849	-2.104569	-0.494929	1.071804
2013-01-04	0.721555	-0.706771	-1.039575	0.271860
2013-01-05	-0.424972	0.567020	0.276232	-1.087401
2013-01-06	-0.673690	0.113648	-1.478427	0.524988

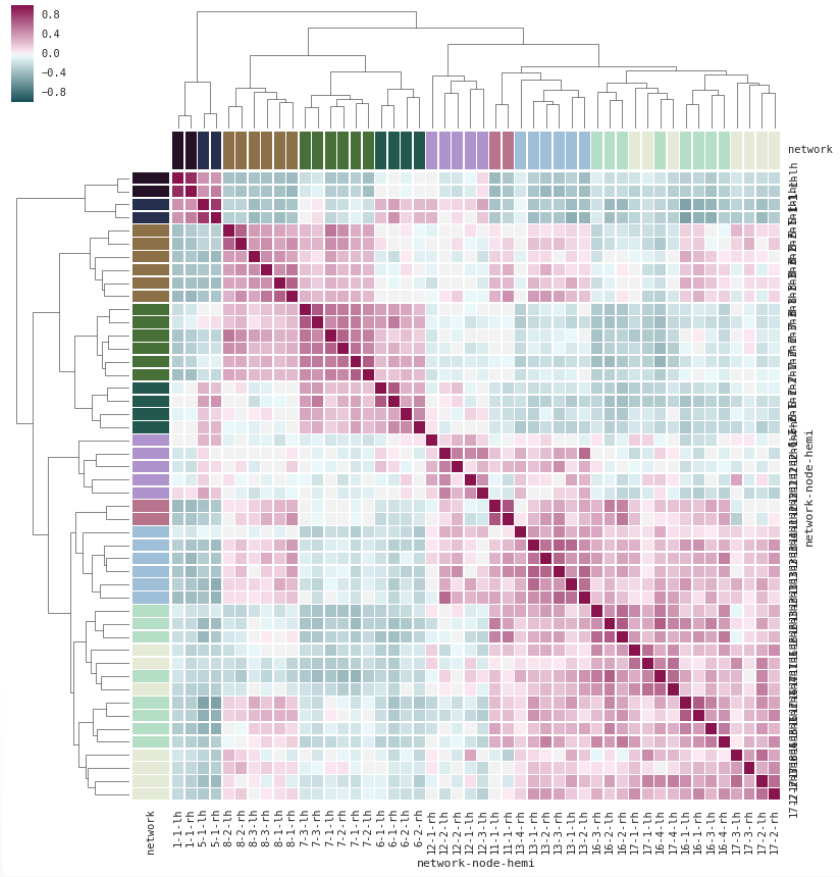
	A	B	C	D	E	F
0	1.0	2013-01-02	1.0	3	test	foo
1	1.0	2013-01-02	1.0	3	train	foo
2	1.0	2013-01-02	1.0	3	test	foo
3	1.0	2013-01-02	1.0	3	train	foo

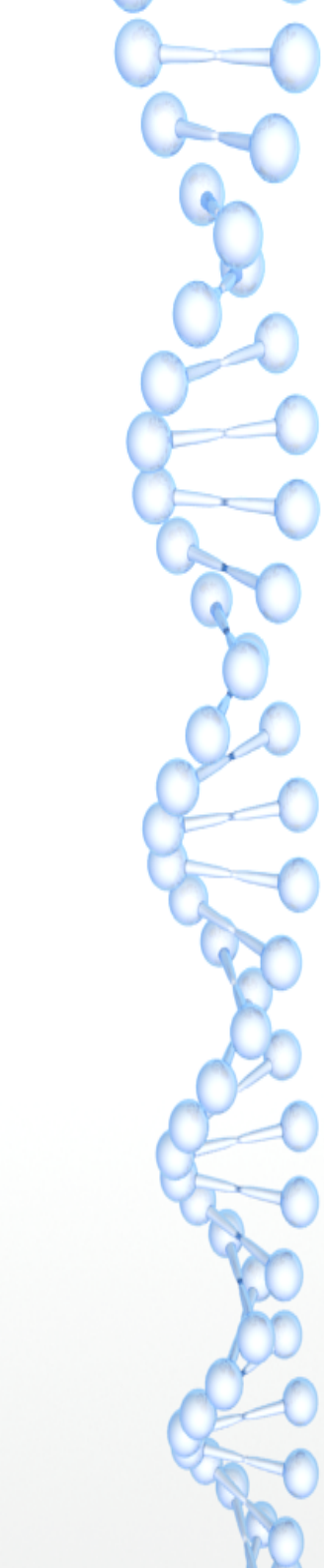
Pandas documentation at:
http://pandas.pydata.org

One more?

Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

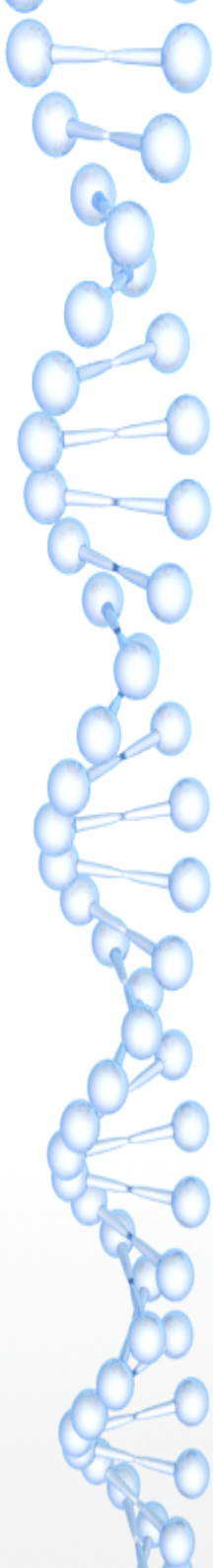






Final recommendations.

- The practice is very important.
- Always read carefully the error messages.
- Do not despair!!!
- There are a lot of tutorials and help on the internet.
- Do not despair!!!
- Have fun!!!



Thank you!!!!