Scipy, Numpy and friends.

Michele Mattioni - mattioni@ebi.ac.uk
CompNeur group.

http://www.ebi.ac.uk/~mattioni/pylab_pres/

EMBL-EBI

Go here:

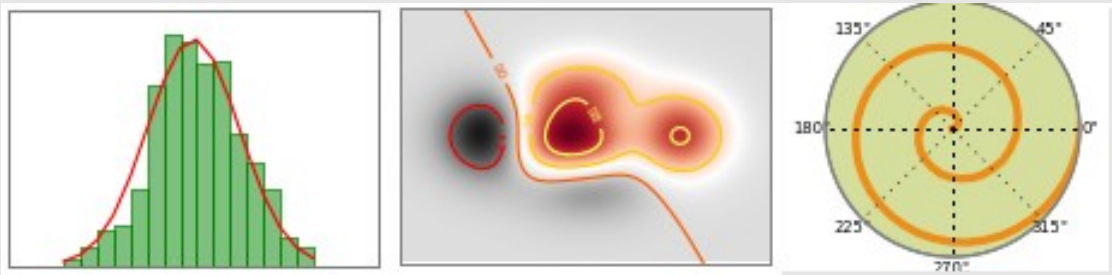
http://www.ebi.ac.uk/~mattioni/pylab_pres/

Scipy and Numpy - http://www.scipy.org

Data structure for mathematical operation with speed in mind.

Collections of high level mathematical operations:
- regression
- interpolation
- optimization
- integration

Matplotlib - http://matplotlib.sourceforge.net/



- Display and plot you data quickly
- Different kind of plot available

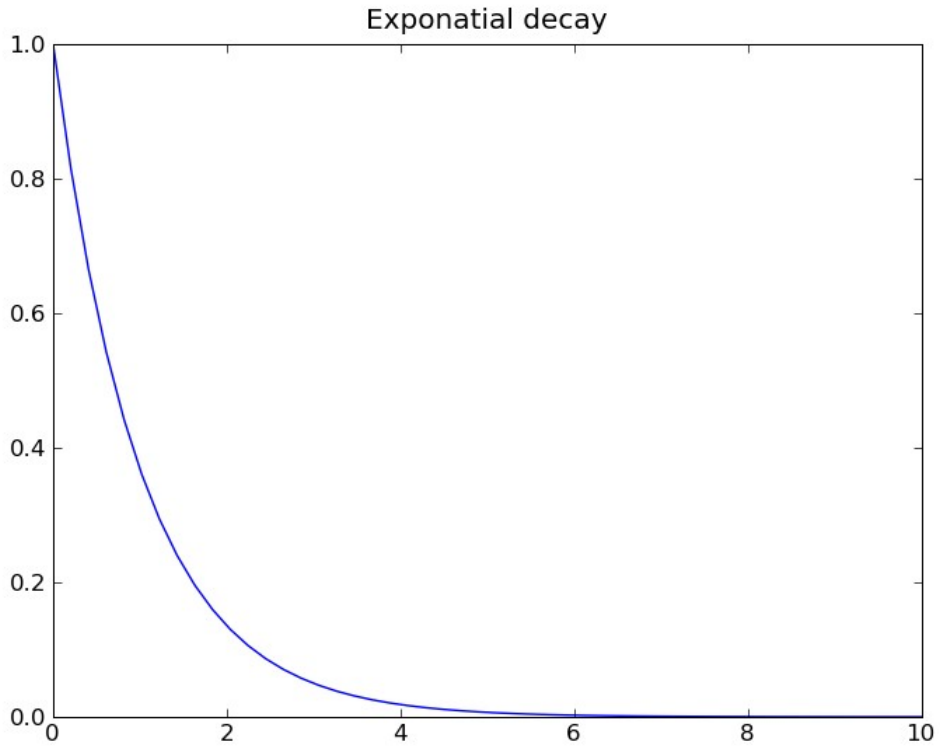
Ipython - http://ipython.scipy.org

IP[y]: IPython

```
mattions@triton:code_ex$ ipython -pylab
/usr/lib/pymodules/python2.6/IPython/Magi
  from sets import Set
Python 2.6.2 (release26-maint, Apr 19 200
Type "copyright", "credits" or "license"

IPython 0.9.1 -- An enhanced Interactive
?         -> Introduction and overview of
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object'. ?obj

  Welcome to pylab, a matplotlib-based Py
  For more information, type 'help(pylab)

In [1]:
```

- Enhanced python console
- Trials and errors encouraged
- Quick access to the help.

Plot of an exponential function




```
1 x = linspace(0,10)
2 plot(x, exp(-x))
```

```
In [1]: x = linspace(0,10)

In [2]: plot(x, exp(-x))
Out[2]: [<matplotlib.lines.Line2D obj

In [3]: plot?
```

Difference between Console and Script

Console

- *Quick to write*
- *Messy to maintain*

```
1 x = linspace(0,10)
2 plot(x, exp(-x))
```

Program

- *Slower to write*
- Easier to maintein

```
1 import numpy as np
2 import scipy as sp
3 import matplotlib.pyplot as plt
4 from numpy import exp
5
6 x = np.linspace(0,10)
7 plt.title('Exponatial decay')
8 plt.plot(x, exp(-x))
9 plt.show()
```

HELP!

Different ways to access help:

Console: name_of_the _function?
Online: http://docs.scipy.org/doc/

Suggested:
Tutorial http://www.scipy.org/Tentative_NumPy_Tutorial

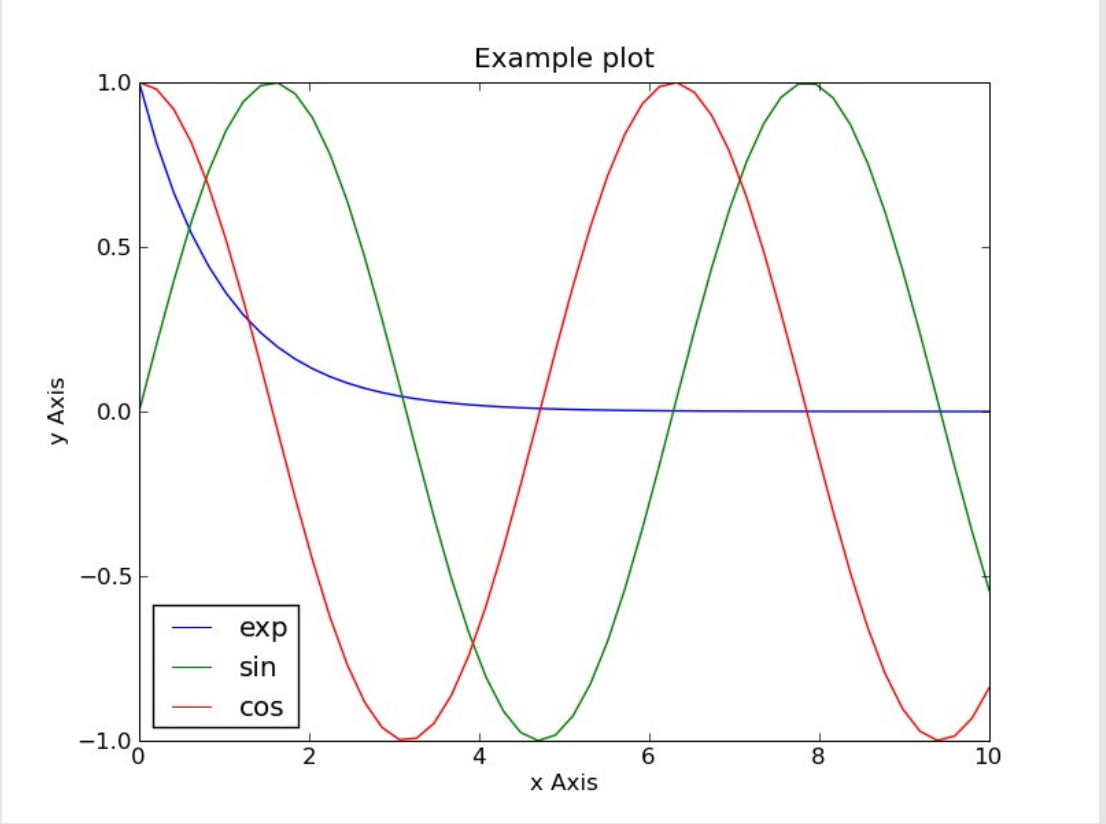
Numpy array

```
# ndarray_ex.py
# This use the console style.
# Do not use this stile on a program.

# Creating an array
x = array([1, 2, 3])
print x*3
# [3 6 9]
print x-1
# [0 1 2]
y = array([10, 20, 30, 40])
print x*y
# [10 40 90]

# Multidimension

z = array([ [1, 2, 3], [10, 20, 30]])
# Can apply the math to the multidimensional
res = z*x + y
print res
#[[ 11,  24,  39],
#[ 20,  60, 120]]

# Indexing
# Row 0:
z[0,:]
# Row 1:
z[1,:]

# Column 0
z[:,0]
# Column 1
z[:,1]
```

```
# ndarray_ex2.py
# This use the console style.
# Do not use this stile on a program.

x = array([1, 2, 3])
y = array([10, 20, 30, 40])
z = array([ [1, 2, 3], [10, 20, 30]])

# The shape
x.shape
y.shape
z.shape

# The dimensions
x.ndim
y.ndim
z.ndim

# number of elements
x.size
y.size
z.size
```

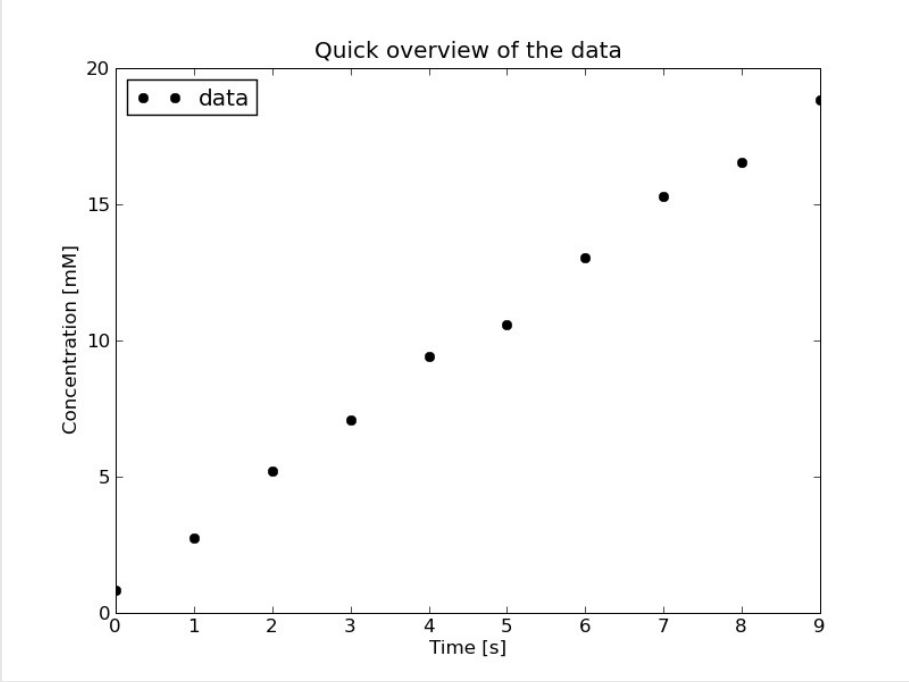
Plotting

```
# plotting.py
# This use the console style.
# Do not use this stile on a program.

x = linspace(0,10)
# Plotting the exp
plot(x, exp(-x), label='exp')
# Plotting the sin
plot(x, sin(x), label='sin')
# Showing the legend
legend()
# Drawing the legend in a better location
legend(loc=0)

# Plotting in another figure
figure()
plot(x, cos(x), label='cos')
legend()

# Plotting in the firt plot
figure(1)
plot(x, cos(x), label='cos')
legend(loc=0)
title('Example plot')
ylabel('y Axis')
xlabel('x Axis')
# saving
savefig('example_plot.png')
```



Importing your data and visualize them

```
1  # displaying_data.py
2  # Proper importing style
3
4  import numpy as np
5  import scipy as sp
6  import matplotlib
7  import matplotlib.mlab
8  import matplotlib.pyplot as plt
9
10 # Importing the data
11
12 try:
13    data = matplotlib.mlab.csv2rec('data.txt')
14    "Data loaded"
15
16 except:
17    print "Bad luck. Where is the file?"
18
19
20 plt.plot(data.x, data.y, 'ko', label='data')
21 plt.title('Quick overview of the data')
22 plt.xlabel('Time [s]')
23 plt.ylabel('Concentration [mM]')
24 plt.legend(loc=0)
25
26 # Line to show the figure
27 plt.show()
```



Fitting the data

```
# fitting_data.py
# Proper importing style

import numpy as np
import scipy as sp
import matplotlib
import matplotlib.mlab
import matplotlib.pyplot as plt

# Importing the data

try:
   data = matplotlib.mlab.csv2rec('data.txt')
   "Data loaded"

except:
   print "Bad luck. Where is the file?"

plt.plot(data.x, data.y, 'ko', label='data')

# Fitting
polycoeffs = np.polyfit(data.x, data.y, 1)
p1 = np.poly1d(polycoeffs)
plt.plot(data.x, p1(data.x), label='fit')

plt.title('Quick overview of the data')
plt.xlabel('Time [s]')
plt.ylabel('Concentration [mM]')
plt.legend(loc=0)
# Line to show the figure
plt.show()
```