

Scipy, Numpy and friends.

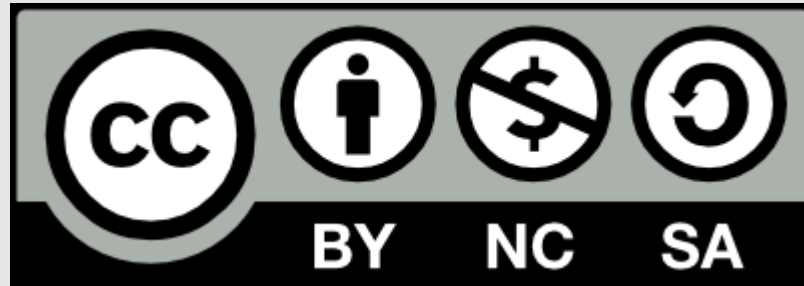
Michele Mattioni - mattioni@ebi.ac.uk
CompNeur group.

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

or

https://github.com/mattions/pylab_pres





Scipy, Numpy and friends by Michele Mattioni is licensed under
CC BY-NC-SA 3.0.

Permissions beyond the scope of this license may be available
at <https://github.com/mattions>.

Go here:

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

or

https://github.com/mattions/pylab_pres

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



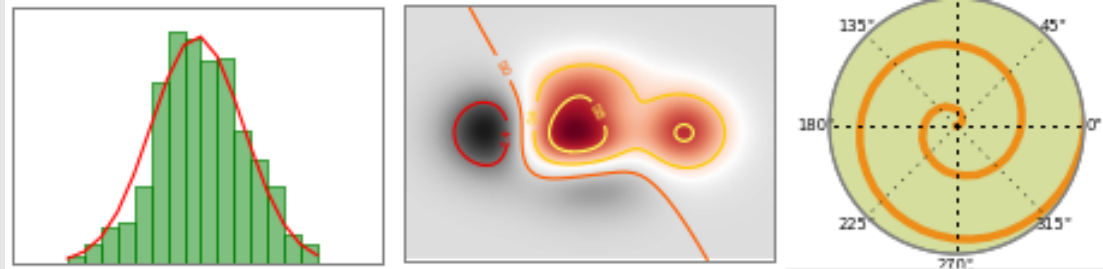
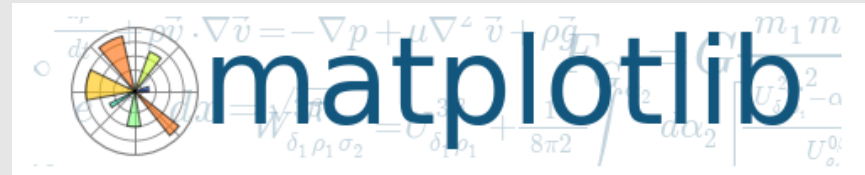
Data structure for
mathematical operation
with speed in mind.



Collections of high level
mathematical operations:

- regression
- interpolation
- optimization
- integration

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



- Display and plot your 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/Magic
from sets import Set
Python 2.6.2 (release26-maint, Apr 19 2006)
Type "copyright", "credits" or "license()"

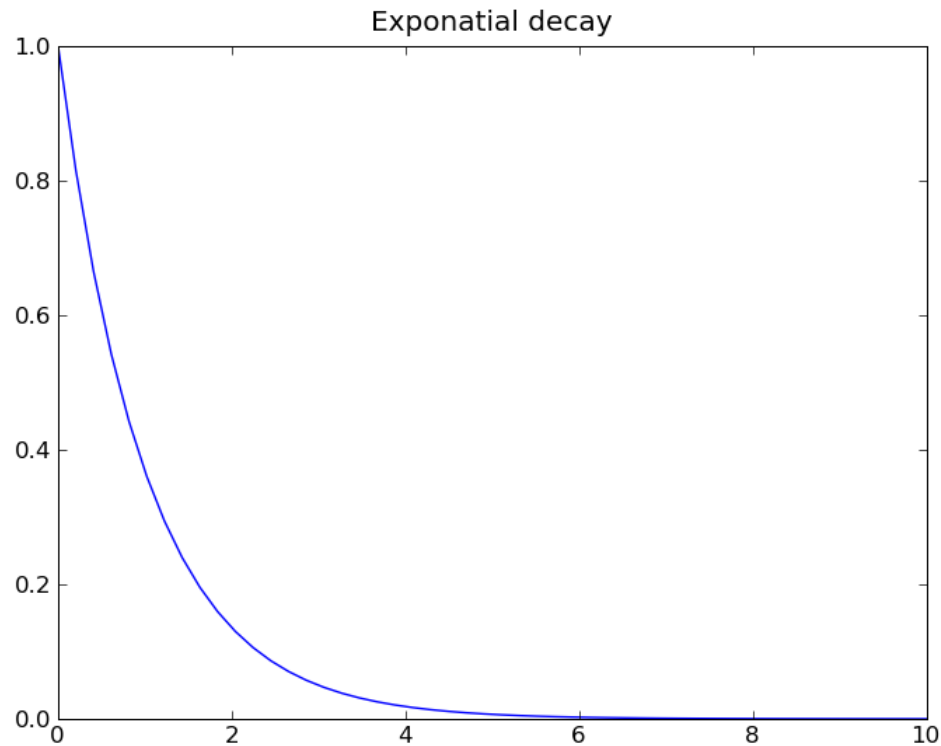
IPython 0.9.1 -- An enhanced Interactive
?          -> Introduction and overview of
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object'. ?obj

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 object>]
```

```
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 maintain*

```
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

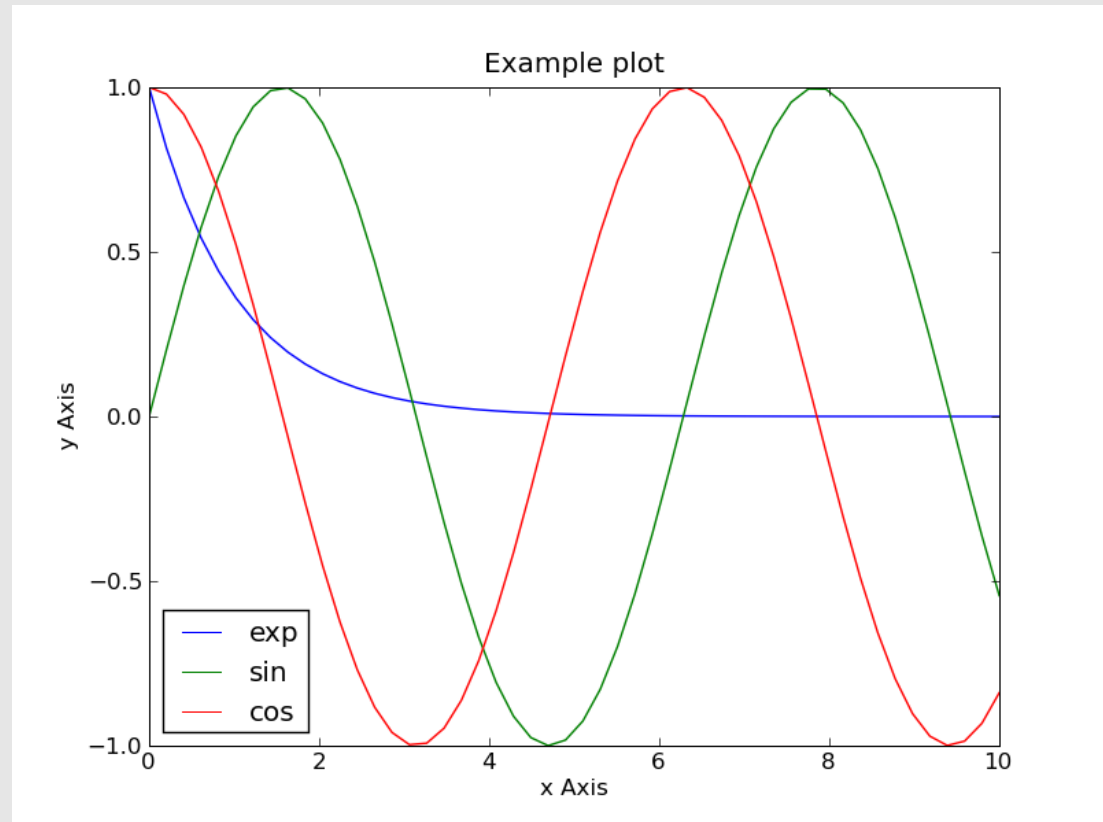
Numpy array

```
1 # ndarray_ex.py
2 # This use the console style.
3 # Do not use this stile on a program.
4
5 # Creating an array
6 x = array([1, 2, 3])
7 print x*3
8 # [3 6 9]
9 print x-1
10 # [0 1 2]
11 y = array([10, 20, 30, 40])
12 print x*y
13 # [10 40 90]
14
15 # Multidimension
16
17 z = array([ [1, 2, 3], [10, 20, 30]])
18 # Can apply the math to the multidimensional
19 res = z*x + y
20 print res
21 #[[ 11,  24,  39],
22 # [ 20,  60, 120]]
23
24 # Indexing
25 # Row 0:
26 z[0,:]
27 # Row 1:
28 z[1,:]
29
30 # Column 0
31 z[:,0]
32 # Column 1
33 z[:,1]
```

```
1 # ndarray_ex2.py
2 # This use the console style.
3 # Do not use this stile on a program.
4
5 x = array([1, 2, 3])
6 y = array([10, 20, 30, 40])
7 z = array([ [1, 2, 3], [10, 20, 30]])
8
9 # The shape
10 x.shape
11 y.shape
12 z.shape
13
14 # The dimensions
15 x.ndim
16 y.ndim
17 z.ndim
18
19 # number of elements
20 x.size
21 y.size
22 z.size
```

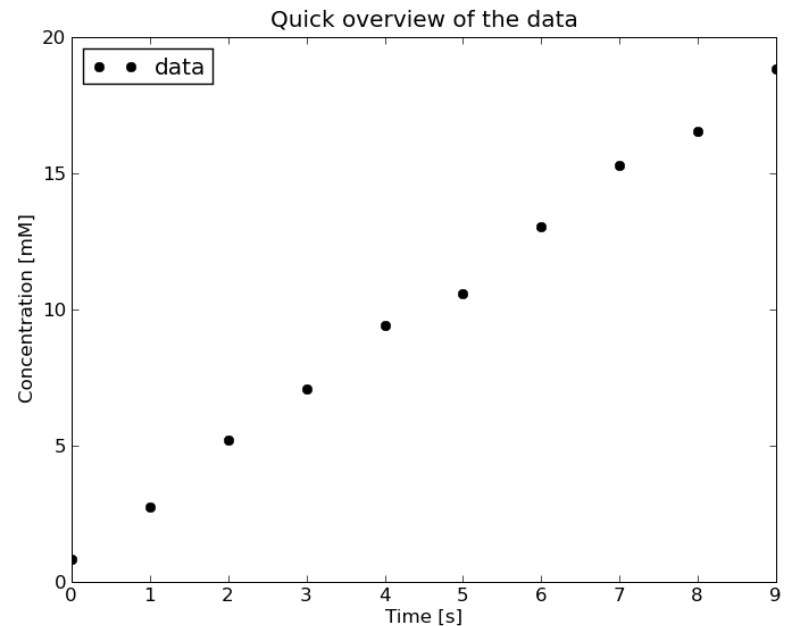
Plotting

```
1 # plotting.py
2 # This use the console style.
3 # Do not use this stile on a program.
4
5 x = linspace(0,10)
6 # Plotting the exp
7 plot(x, exp(-x), label='exp')
8 # Plotting the sin
9 plot(x, sin(x), label='sin')
10 # Showing the legend
11 legend()
12 # Drawing the legend in a better location
13 legend(loc=0)
14
15 # Plotting in another figure
16 figure()
17 plot(x, cos(x), label='cos')
18 legend()
19
20 # Plotting in the first plot
21 figure(1)
22 plot(x, cos(x), label='cos')
23 legend(loc=0)
24 title('Example plot')
25 ylabel('y Axis')
26 xlabel('x Axis')
27 # saving
28 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

```
1 # fitting_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 plt.plot(data.x, data.y, 'ko', label='data')
20
21 # Fitting
22 polycoeffs = np.polyfit(data.x, data.y, 1)
23 p1 = np.poly1d(polycoeffs)
24 plt.plot(data.x, p1(data.x), label='fit')
25
26 plt.title('Quick overview of the data')
27 plt.xlabel('Time [s]')
28 plt.ylabel('Concentration [mM]')
29 plt.legend(loc=0)
30 # Line to show the figure
31 plt.show()
```

