

Pylab (Matplotlib) Introduction

Molecular Statistics

2014

1 Introduction

Pylab is a module for Python used for plotting, which utilizes another module called Matplotlib. It is very useful for plotting all kinds of data and is extremely customizable. See the gallery matplotlib.org/gallery.html, or webloria.loria.fr/~rougier/teaching/matplotlib for examples.

In this short introduction we will go through small useful examples with provided code. For more advanced examples (like 3D plots), see the links above or do a Google-search.

All the examples below will be using the python package **pylab**, and must thus be included in your script. This is done by including

```
1 import pylab
```

in the head of the `.py`-file.

To illustrate the result of a plot/graph, you can either show it in a new window or save the plot as an image.

```
1 # Show the result in a new window
2 pylab.show()
3
4 # Save the result in a file
5 pylab.savefig('this_figure.png')
```

The `savefig()` method can save to `.png`, `.eps`, `.svg`, as well as `.pdf`.

When creating multiple plots in the same script, remember to *clear figure* after each plot, using

```
1 pylab.clf()
```

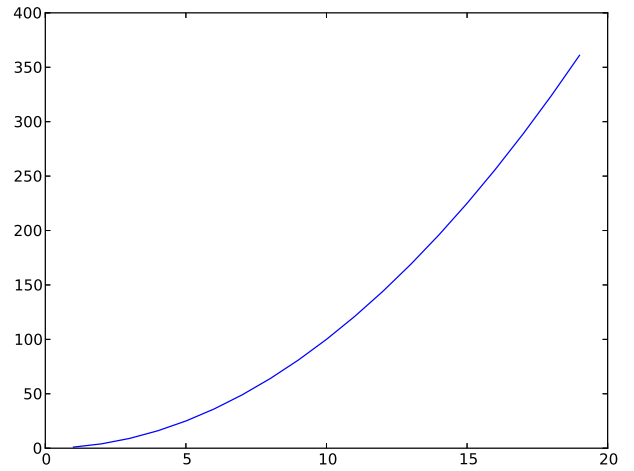
which will erase all previous work done in pylab.

2 Examples

2.1 XY

A simple example is to plot x and y coordinates based on two simple python lists.

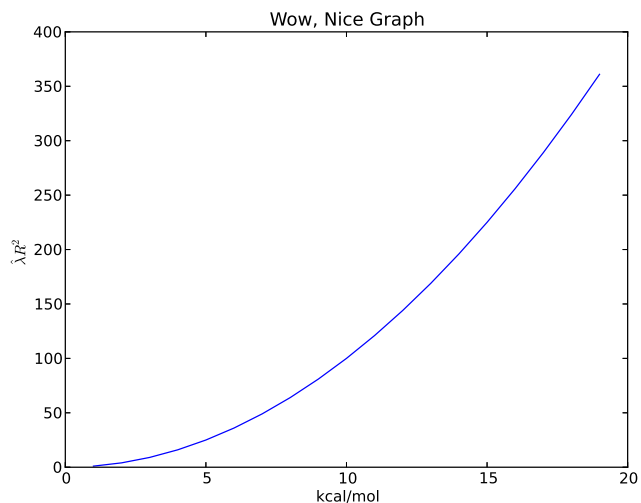
```
1 import pylab
2
3 x = range(1, 20)
4 y = [i**2 for i in x]
5
6 # Plot x- and y coordinates
7 pylab.plot(x, y)
8 pylab.savefig('xy_figure.png')
```



2.2 Titles and labels

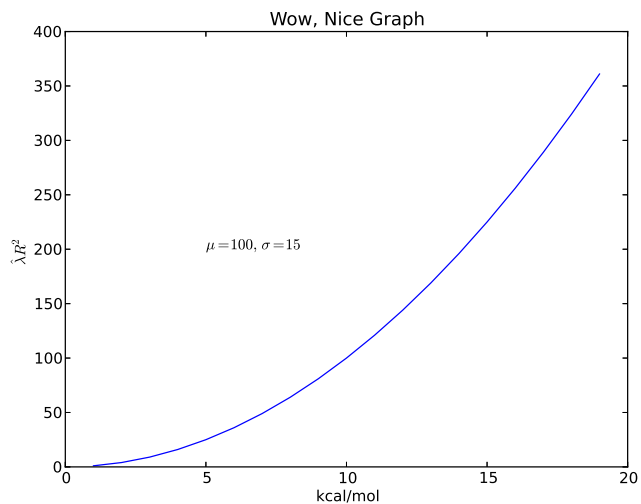
However a graph is worth nothing without title and axis-labels, which can be included by the following syntax. It is also possible to include latex code in the strings.

```
1 import pylab
2
3 x = range(1, 20)
4 y = [i**2 for i in x]
5
6 # Plot x- and y coordinates
7 pylab.plot(x, y)
8
9 # Set title
10 pylab.title("Wow, Nice graph")
11
12 # Axis
13 pylab.xlabel('kcal/mol')
14 pylab.ylabel('$\hat{\lambda} R^2$')
15
16 pylab.savefig('figure_name.png')
```



Text can also be inserted into the figure, for example

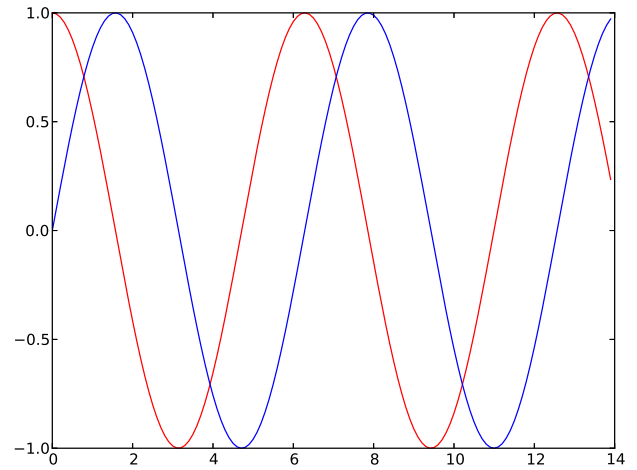
```
1 import pylab
2
3 x = range(1, 20)
4 y = [i**2 for i in x]
5
6 # Plot x- and y coordinates
7 pylab.plot(x, y)
8
9 # Set title
10 pylab.title("Wow, Nice graph")
11
12 # Axis
13 pylab.xlabel('kcal/mol')
14 pylab.ylabel('$\hat{\lambda} R^2$')
15
16 # Text
17 pylab.text(5, 200, r'$\mu=100, \sigma=15$')
18
19 pylab.savefig('figure_name.png')
```



2.3 Multiple lines

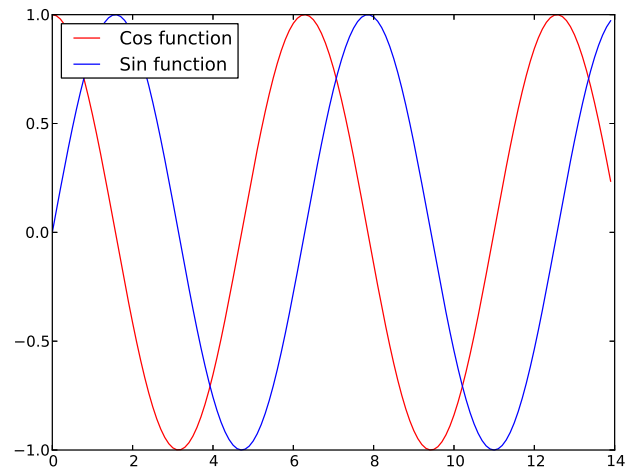
Plotting multiple datasets in the same figure can be done by choosing a color and a line style for the plot. See appendix A for a full list of colors and line styles.

```
1 import pylab
2 import math
3
4 x = numpy.arange(0, 14, 0.1)
5 y_cos = [math.cos(i) for i in x]
6 y_sin = [math.sin(i) for i in x]
7
8 # Insert x and y coordinates
9 # with a red and blue line
10 pylab.plot(x, y_cos, 'r-')
11 pylab.plot(x, y_sin, 'b-')
12
13 pylab.savefig('figure_sincos1.png')
```



Having labels/legends on the plots can be fairly important. This is done by assigning each plot a label, and inserting them in the figure by using the legend command. Note that the location of the legend box can be set by changing the string location.

```
1 import pylab
2 import math
3
4 x = numpy.arange(0, 14, 0.1)
5 y_cos = [math.cos(i) for i in x]
6 y_sin = [math.sin(i) for i in x]
7
8 pylab.plot(x, y_cos, 'r-', label="Cos
9         function")
10 pylab.plot(x, y_sin, 'b-', label="Sin
11         function")
12
13 pylab.legend(loc='upper left')
14
15 pylab.savefig('figure_sincos2.png')
```



A Plot styling

A.1 Colors

b	blue
g	green
r	red
c	cyan
m	magenta
y	yellow
k	black
w	white

A.2 Line styles

0	tickleft
1	tickright
2	tickup
3	tickdown
4	caretleft
D	diamond
6	caretup
7	caretdown
s	square
—	vline
x	x
5	caretright
-	hline
^	triangle up
d	thin diamond
h	hexagon1
+	plus
*	star
,	pixel
o	circle
.	point
'1'	tri down
p	pentagon
'3'	tri left
'2'	tri up
'4'	tri right
H	hexagon2
v	triangle down
'8'	octagon
<	triangle left
>	triangle right