

```
1 import numpy, pylab
2 from NeuroTools.parameters import ParameterSpace

4 from NeuroTools.parameters import ParameterRange
5 from NeuroTools.sandbox import make_name
6 # import function that generates correlated spikes trains from a ParameterSet
7 # returns: cc - correlation coefficient, time_axis_cc - time axis for cross-correlation (for plotting), corrcoef -
8 # correlation coefficient between the two SpikeTrains
9 from examples_neurotools_parameters_tools import calc_cc

    creating a ParameterSpace
12 p = ParameterSpace({})

    adding fixed parameters
14 p.nu = 20. #! rate [Hz]
15 p.duration = 10000.
16
17 # adding ParameterRanges
18 p.c = ParameterRange([0.0,0.01,0.1,0.5])
19 p.jitter = ParameterRange([0.0,0.5,1.0,1.5])
20
21 # calculation of the ParameterSpace dimension and the labels of the parameters
22 # containing a range
23 dims, labels = p.parameter_space_dimension_labels()
24 print "dimensions: ", dims

    dimensions: [4, 4]
26 print ' labels: ', labels

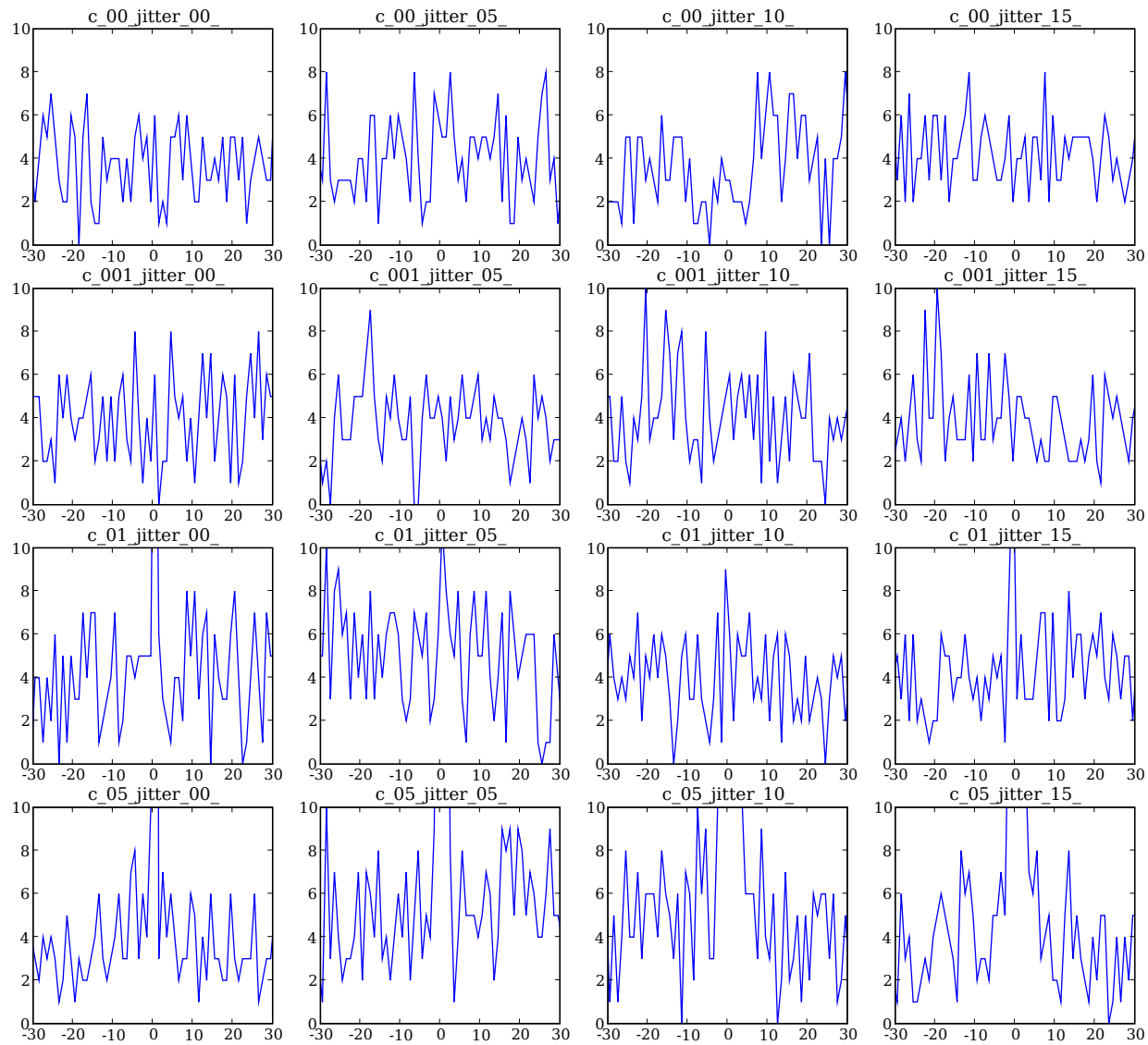
    labels: ['c', 'jitter']

26 # creating a results array, with the dimensions of the ParameterSpace
27 corrcoef_results = numpy.empty(dims)
28 print corrcoef_results.shape

    (4, 4)

31 pylab.rcParams['figure.figsize'] = [16,14]
32 pylab.figure()
33
34
```

```
35 # scanning the ParameterSpace
36 for experiment in p.iter_inner():
37     # calculation of the index in the space
38     index = p.parameter_space_index(experiment)
39     # performing the experiment
40     cc,time_axis_cc, corrcoef = calc_cc(experiment)
41     corrcoef_results[index] = corrcoef
42     # plotting the cc's
43     subplot_index = (dims[1]*index[0])+index[1]
44     pylab.subplot(dims[0],dims[1],subplot_index+1)
45     pylab.plot(time_axis_cc,cc)
46     pylab.title(make_name(experiment,p.range_keys()))
47     pylab.xlim(-30,30.)
48     pylab.ylim(0,10.)
49
50 pylab.show()
```



```
50 # plot the results
51 pylab.matshow(corrcoef_results)
52 pylab.xticks(numpy.arange(0.5,dims[1]+0.5,1.0),[str(i) for i in p.jitter._values])
53 pylab.yticks(numpy.arange(0.5,dims[0]+0.5,1.0),[str(i) for i in p.c._values])
54 pylab.xlim(0,dims[1])
55 pylab.ylim(dims[0],0)
56 pylab.xlabel('jitter (ms)')
57 pylab.ylabel('correlation')
58 ax = pylab.colorbar()
59 ax.set_label('correlation')
60 pylab.draw()
61
62 pylab.show()
```

