**Working with the set of stochastic simulations**

In the assessed practical you were set the task of loading and analysing a set of 100 runs of the stochastic model, that all started with the same initial conditions.

On the following pages you can find some hints and example code for working with this data.

Page 2: Hint on how to load observations from “output” array.

Page 3: Hint on how to create array of LacI observations ready for plotting / averaging

Page 4. Hint on how to measure height and period by using peak finder on a set of LacI observations

Page 2

Hint on how to load observations from “output” array.

1. output=load('repressilator\_output.pickle')
2. **print** "# runs in output:",len(output)
3. **print** "# observations in each run",len(output[0])
5. # load the observations from run 0
6. s\_obs0=output[0]
8. # load the observations from run 1
9. s\_obs1=output[0]
11. # unpack the observations from run 0
12. p\_LacI\_obs0  = s\_obs0[:,0]
13. p\_TetR\_obs0  = s\_obs0[:,1]
14. p\_CI\_obs0    = s\_obs0[:,2]
15. m\_LacI\_obs0  = s\_obs0[:,3]
16. m\_TetR\_obs0  = s\_obs0[:,4]
17. m\_CI\_obs0    = s\_obs0[:,5]
19. # plot example results
20. fig=plt.figure()
21. ax=fig.add\_subplot(1,1,1)
22. ax.plot(t\_obs/60, p\_LacI\_obs0, 'b-')
23. fig.show()

Page 3

Hint on how to create array of LacI observations ready for plotting / averaging

1. # loop over all runs stored in output and create array of LacI observations
2. LacI\_obs\_runs=[]
3. **for** s\_obs **in** output:
4. LacI\_obs=s\_obs[:,0]
5. LacI\_obs\_runs.append(LacI\_obs)
7. # transpose so all runs can be plotted
8. LacI\_obs\_runs=np.transpose(LacI\_obs\_runs)
10. # plot trajectories
11. fig2=plt.figure()
12. ax2=fig2.add\_subplot(1,1,1)
13. ax2.plot(t\_obs/60, LacI\_obs\_runs,'-', alpha=0.4)
14. fig2.show()

Page 4.

Hint on how to measure height and period by using peak finder on a set of LacI observations

1. heights=[]
2. periods=[]
3. # find peaks in p\_LacI\_obs0
4. peaks=peak\_finder(t\_obs, p\_LacI\_obs, 500)
5. **for** i **in** range(len(peaks)):
6. peak\_time = peaks[i][0]
7. peak\_height = peaks[i][1]
8. **print** "found peak at time {:.0f} with height {:.0f}".format(peak\_time, peak\_height)
9. heights.append(peak\_height)
10. **if** i>=1:
11. period= peaks[i][0]- peaks[i-1][0]
12. periods.append(period)
13. **print** heights
14. **print** periods

To draw a distribution of all the recorded heights/periods you need to use loop e.g. for each LacI\_obs=s\_obs[:,0] stored in output (so you collect data from all 100 runs).