**Coding the stochastic simulation of the Repressilator**

To complete the code in repressilator\_stoch.py we need to complete the section of the Gillespie algorithm that updates the number of each species according to what reaction has taken place.

You can see a completed example in the model answer for the stochastic simulation of gene expression:

1. #types=['on','off','t','p','dmRNA','dp']
2. **if** event\_type=="on":
3. g\_on+=1
4. g\_off+=-1
5. **elif** event\_type=="off":
6. g\_off+=1
7. g\_on+=-1
8. **elif** event\_type=="t":
9. mRNA+=1
10. **elif** event\_type=="p":
11. p+=1
12. **elif** event\_type=="dmRNA":
13. mRNA+=-1
14. **elif** event\_type=="dp":
15. p+=-1
16. **else**:
17. **print** "error unknown event type!!"

In this case we have the following event types:

1. types=["m\_LacI\_prod",  # one copy of m\_LacI produced
2. "m\_TetR\_prod",  # one copy of m\_TetR produced
3. "m\_CI\_prod",    # one copy of m\_CI   produced
4. "m\_LacI\_loss",  # one copy of m\_LacI lost
5. "m\_TetR\_loss",  # one copy of m\_TetR lost
6. "m\_CI\_loss",    # one copy of m\_CI   lost
7. "p\_LacI\_prod",  # one copy of p\_LacI produced
8. "p\_TetR\_prod",  # one copy of p\_TetR produced
9. "p\_CI\_prod",    # one copy of p\_CI   produced
10. "p\_LacI\_loss",  # one copy of p\_LacI lost
11. "p\_TetR\_loss",  # one copy of p\_TetR lost
12. "p\_CI\_loss",    # one copy of p\_CI   lost
13. "m\_GFP\_prod",   # one copy of m\_GFP  produced
14. "m\_GFP\_loss",   # one copy of m\_GFP  lost
15. "p\_GFP\_prod",   # one copy of p\_GFP  produced
16. "p\_GFP\_loss",   # one copy of p\_GFP  lost
17. ]

Therefore we need to add lines like:

1. **if** event\_type=="m\_LacI\_prod":
2. m\_LacI+=1

and

1. **elif** event\_type=="m\_LacI\_loss":
2. m\_LacI+=-1

When completed you should find the stochastic simulation produces a timeseries that is a “noisy” version of the deterministic model.