```
In [1]:  import numpy as np
         from scipy.integrate import odeint
         from scipy.optimize import curve_fit
         from scipy.optimize import differential_evolution
         %matplotlib inline
         import matplotlib.pyplot as plt
         from matplotlib.backends.backend_pdf import PdfPages
         import pandas as pd
```

```
In [2]:  from endpoint_maker import sqres
         from endpoint_maker import func
         from endpoint_maker import t
         from endpoint_maker import new
         from endpoint_maker import lightdata
         from endpoint_maker import rawdata
```

```
In [3]:  if __name__ == '__main__':
             def initparams():
                 bounds = ([0.00001,10000],[0,4],[0.00001,1000],[0.00001,1000],[0.00001
                 #bounds = ([0.00001,1000],[0.00001,1000],[0.00001,10000])
                 result = differential_evolution(sqres,bounds,maxiter=100,popsize=20,po
                 return result
```

```
In [4]:  initialp = initparams()
         print(initialp)
```

```
 message: Maximum number of iterations has been exceeded.
 success: False
     fun: 2.85422416023694
       x: [ 8.206e+03  1.455e+00  5.024e+02  3.368e+00  1.281e+01]
     nit: 100
    nfev: 10100
```

```python
In [5]: def tester(t,Kd,n,d2,k2,k3,i):
        #def tester(t,d2,k2,k3,i):

            inivalues = [1,0,0,0,0,0,0]
            arrayvalues = np.asarray([])

            #for i in range(len(lightdata[:,0])):
            def I(t):
                tindex = t/5
                if tindex > 12241:
                    tindex = 12240
                return lightdata[i][int(tindex)]

            def odes(z,t,Kd,n,d2,k2,k3):
            #def odes(z,t,d2,k2,k3):
                Pu,Pb,Pa,mRNA,mCherry1,mCherry2,mCherry3 = z
                d1 = 0.019905
                k1 = 0.08299
                #Kd = 90.41
                #n = 0.964487
                #d2= 486.67
                #k2= 6.597
                #k3= 0.0539


                d3 = 0.000077
                k4 = 1.25
                d4 = 0.000031
                k5 = 0.00283
                k6 = 0.00283

                Pu = z[0]
                Pb = z[1]
                Pa = z[2]
                mRNA = z[3]
                mCherry1 = z[4]
                mCherry2 = z[5]
                mCherry3 = z[6]

                dPudt = d1*Pb - k1*I(t)**n/(Kd**n+I(t)**n)*Pu
                dPbdt = k1*I(t)**n/(Kd**n+I(t)**n)*Pu - k2*Pb - d1*Pb + d2*Pa
                dPadt = k2*Pb - d2*Pa
                dmRNAdt = k3*I(t)**n/(Kd**n+I(t)**n)*Pa - d3*mRNA
                dmCherry1dt = k4*mRNA-(d4 + k5)*mCherry1
                dmCherry2dt = k5*mCherry1-(d4+k6)*mCherry2
                dmCherry3dt = k6*mCherry2 - d4*mCherry3


                return [dPudt,dPbdt,dPadt,dmRNAdt,dmCherry1dt,dmCherry2dt,dmCherry3dt]

            solver = odeint(odes,inivalues,t,args = (Kd,n,d2,k2,k3),hmax=0.1)
            #solver = odeint(odes,inivalues,t,args = (d2,k2,k3),hmax=0.1)

            mCherryout = solver[:,6]
            #mCherryout = mCherryout[0:24480:240]
            return mCherryout
```

In [6]:
```python
print(initialp.x)
```

```
[8.20644511e+03 1.45480016e+00 5.02379238e+02 3.36849925e+00
 1.28050002e+01]
```

In [7]:
```python
popt, covt = curve_fit(func,t,new,initialp.x,maxfev=1000000)

#popt, covt = curve_fit(func,t,newdata,initialp.x, maxfev=10000000, bounds=((0
#popt, covt = curve_fit(func,t,newdata,initialp.x, maxfev=1000000)

Kd,n,d2,k2,k3 = popt
print('Kd=',Kd,'n=',n,'d2=',d2,'k2=',k2,'k3=',k3)

#d3,k4,d4,k5,k6 = popt
#print('d3=',d3,'k4=',k4,'d4=',d4,'k5=',k5,'k6=',k6)

#d2,k2,k3 = popt
#print('d2=',d2,'k2=',k2,'k3=',k3)
```

```
Kd= 8064.753755175012 n= 1.4587228822120382 d2= 508.7708723008923 k2= 3.54674
4574534316 k3= 13.991115085987797
```

In [8]:
```python
 1  import sys
 2  import numpy
 3  #params = [1,1,1,1,1,1,1]
 4  params = [1,1,1,1,1]
 5  numpy.set_printoptions(threshold=10)
 6  model1 = np.asarray(func(t,Kd,n,d2,k2,k3))
 7  #model1 = np.asarray(func(t,d2,k2,k3))
 8  print(len(model1))
 9  #print(model1)
10
11  #a,b,c,d,e,f,g = params
12  a,b,c,d,e = params
13
14  ydata = np.asarray(new)
15  print(len(ydata))
16
17  ssr = np.sum((ydata-model1)**2)
18  #ssr2 = np.sum((ydata-model2)**2)
19  sst = np.sum((ydata - np.mean(ydata))**2)
20  R2 = 1 - ssr/sst
21  #R2_2 = 1 - ssr2/sst
22
23  print('R2 is: ', R2)
24  #print(R2_2)
```

```
102
102
R2 is:  -0.13390047420505047
```

In [9]:
```python
pp = PdfPages('multipage.pdf')
ydata = np.asarray(new)

#condition = [1,2,3,4,5,6,7,8,9]
for i in range(2):
    model = tester(t,Kd,n,d2,k2,k3,i)
    #model = tester(t,d2,k2,k3,i)

    #print(model)
    #a,b,c,d,e,f,g= params
    a,b,c,d,e= params

    plt.plot(t,model,'.', label = 'model')
    #print(model)
    #t = np.linspace(0,34800, num=6961)
    #raw = rawdata[0:13920:240]
    plt.plot(t,raw[i],'.',label = 'data')
    #print(rawdata[i])
    plt.legend()
    pp.savefig()
    plt.show()

pp.close
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[9], line 17
     13 plt.plot(t,model,'.', label = 'model')
     14 #print(model)
     15 #t = np.linspace(0,34800, num=6961)
     16 #raw = rawdata[0:13920:240]
---> 17 plt.plot(t,raw[i],'.',label = 'data')
     18 #print(rawdata[i])
     19 plt.legend()

NameError: name 'raw' is not defined
```
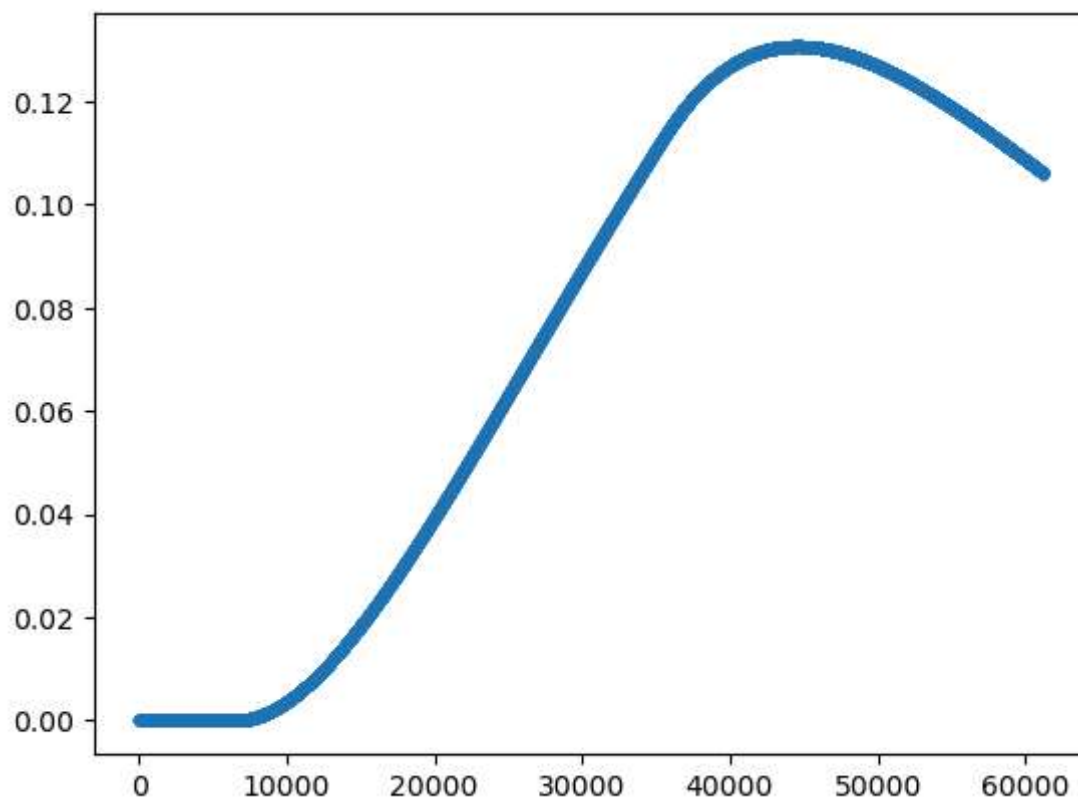
In [ ]: