

Simulación estocástica con algoritmo de Gillespie

Punto a

```
In [1]: %pylab inline
```

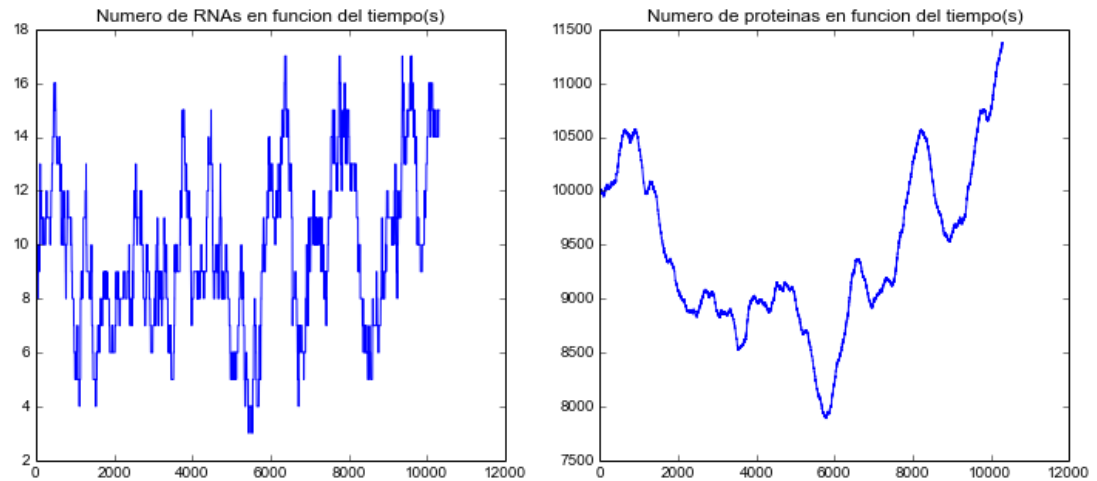
Populating the interactive namespace from numpy and matplotlib

El algoritmo de Gillespie no tiene un dt fijo, sino que hace lanzamientos aleatorios para decidir cuándo será el próximo evento, y luego decide con otro lanzamiento aleatorio a qué evento corresponde. Suponiendo 1000 datos con un dt cercano a 1s, significa que todo el experimento dure alrededor de 1000s. Con las probabilidades dadas para nuestro experimento, el tiempo promedio entre eventos es de 0.09 segundos (para cuando está en equilibrio), lo que nos obliga a correr un mínimo de 11000 eventos aproximadamente para asegurar 1000s de experimento. Se decide iniciar en los valores estacionarios de RNAs y proteínas para que ese número sea independiente del número de eventos.

```
In [2]: gamma=1./(60*5.)
gamma_p=1./(60*30.)
k_p=1000.*gamma_p
k=10.*gamma
steps=110000
tiempo=linspace(0.,steps,steps+1)
r,p=zeros(size(tiempo)),zeros(size(tiempo))
r[0],p[0],tiempo[0]=10,10000,0.
dt=[]
for i in range(size(tiempo)-1):
    T_cr=k
    T_des=gamma*r[i]
    T3=k_p*r[i]
    T4=gamma_p*p[i]
    T_tot=T_cr+T_des+T3+T4
    t=-1./T_tot*(log(random.random()))
    dt.append(t)
    tiempo[i+1]=tiempo[i]+t
    x=random.random()
    if x<(T_cr/T_tot):
        r[i+1]=r[i]+1
        p[i+1]=p[i]
    elif x<(T_cr+T_des)/T_tot:
        r[i+1]=r[i]-1
        p[i+1]=p[i]
    elif x<(T_cr+T_des+T3)/T_tot:
        r[i+1]=r[i]
        p[i+1]=p[i]+1
    else:
        r[i+1]=r[i]
        p[i+1]=p[i]-1
```

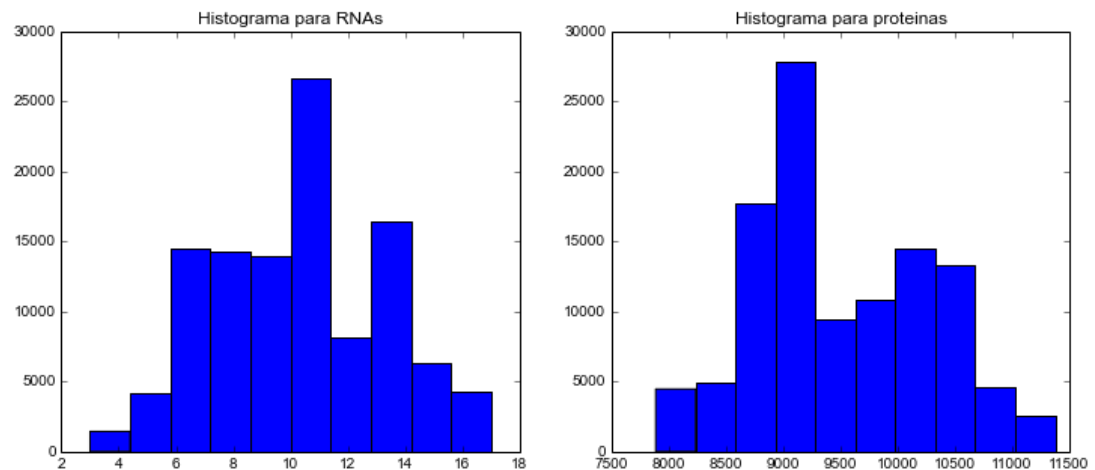
```
In [3]: subplots(1,2,figsize=(12,5))
subplot(1,2,1)
title('Numero de RNAs en funcion del tiempo(s)')
plot(tiempo,r)
subplot(1,2,2)
title('Numero de proteinas en funcion del tiempo(s)')
plot(tiempo,p)
```

Out[3]: [<matplotlib.lines.Line2D at 0x7f7d55335e10>]



```
In [10]: subplots(1,2,figsize=(12,5))
binis=10
subplot(1,2,1)
title('Histograma para RNAs')
hist(r,bins=binis)
subplot(1,2,2)
title('Histograma para proteinas')
hist(p,bins=binis)
```

```
Out[10]: (array([ 4465.,  4887., 17679., 27867.,  9396., 10786., 14473.,
        13303.,  4577., 2568.]),
 array([ 7888.,  8236.7,  8585.4,  8934.1,  9282.8,  9631.5,
        9980.2, 10328.9, 10677.6, 11026.3, 11375. ]),
 <a list of 10 Patch objects>)
```



```
In [5]: print "El numero promedio de proteinas es de ",average(p)," y su desviacion es
tandar es",std(p)
print "El numero promedio de RNAs es de ",average(r)," y su desviacion estanda
r es",std(r)
print "El numero promedio entre eventos es de ",average(dt)," y su desviacion
estandar es",std(dt)
```

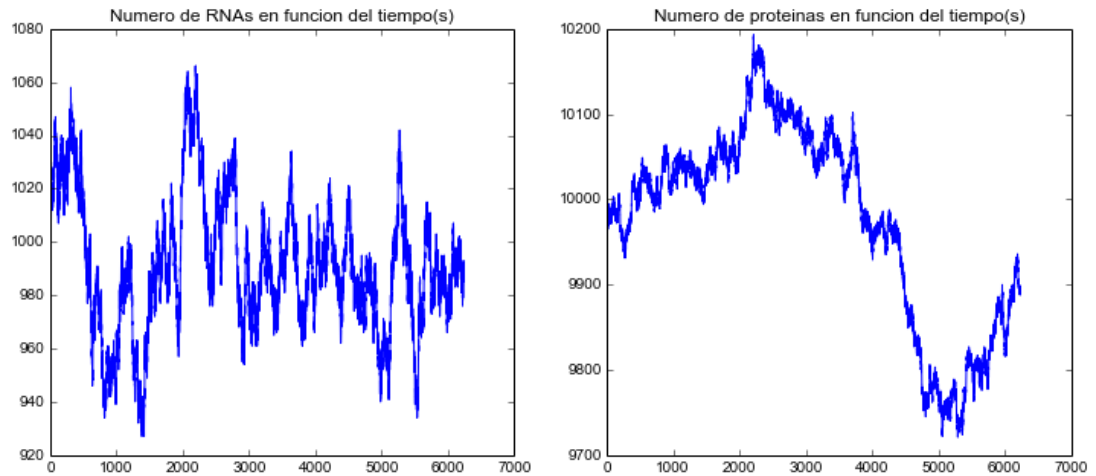
```
El numero promedio de proteinas es de  9495.08172653  y su desviacion estandar
es 761.93907201
El numero promedio de RNAs es de  10.2359524004  y su desviacion estandar es 2.
9753960813
El numero promedio entre eventos es de  0.0935838065134  y su desviacion stand
ar es 0.096430113476
```

Punto b

```
In [6]: gamma2=1./(60*5.)
gamma_p2=1./(60*30.)
k_p2=10.*gamma_p
k2=1000.*gamma
steps2=110000
tiempo2=linspace(0.,steps2,steps2+1)
r2,p2=zeros(size(tiempo2)),zeros(size(tiempo2))
r2[0],p2[0],tiempo2[0]=1000,10000,0.
dt2=[]
for i in range(size(tiempo2)-1):
    T_cr2=k2
    T_des2=gamma2*r2[i]
    T32=k_p2*r2[i]
    T42=gamma_p2*p2[i]
    T_tot2=T_cr2+T_des2+T32+T42
    t2=-1./T_tot2*(log(random.random()))
    dt2.append(t2)
    tiempo2[i+1]=tiempo2[i]+t2
    x2=random.random()
    if x2<(T_cr2/T_tot2):
        r2[i+1]=r2[i]+1
        p2[i+1]=p2[i]
    elif x2<(T_cr2+T_des2)/T_tot2:
        r2[i+1]=r2[i]-1
        p2[i+1]=p2[i]
    elif x2<(T_cr2+T_des2+T32)/T_tot2:
        r2[i+1]=r2[i]
        p2[i+1]=p2[i]+1
    else:
        r2[i+1]=r2[i]
        p2[i+1]=p2[i]-1
```

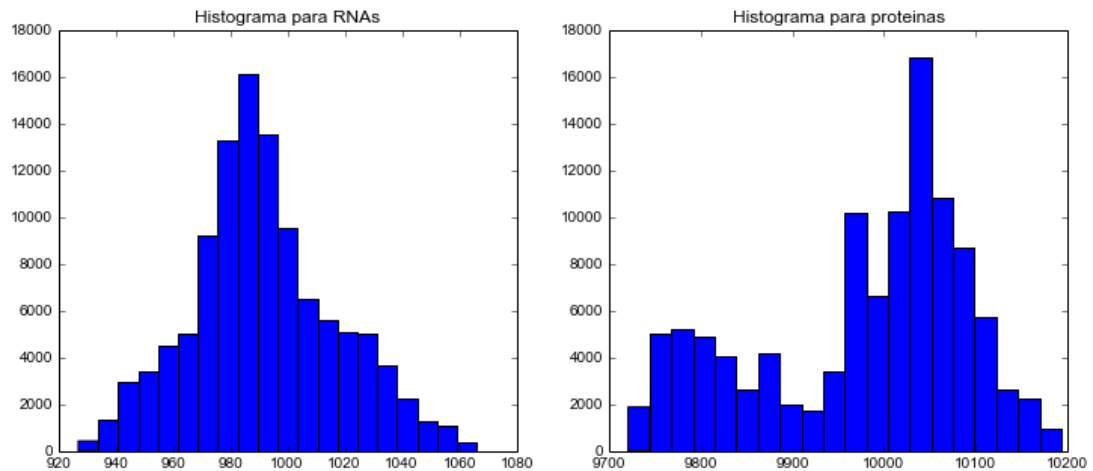
```
In [7]: subplots(1,2,figsize=(12,5))
subplot(1,2,1)
title('Numero de RNAs en funcion del tiempo(s)')
plot(tiempo2,r2)
subplot(1,2,2)
title('Numero de proteinas en funcion del tiempo(s)')
plot(tiempo2,p2)
```

Out[7]: [<matplotlib.lines.Line2D at 0x7f7d54d6ff10>]



```
In [8]: subplots(1,2,figsize=(12,5))
        bins=20
        subplot(1,2,1)
        title('Histograma para RNAs')
        hist(r2,bins=bins)
        subplot(1,2,2)
        title('Histograma para proteinas')
        hist(p2,bins=bins)
```

```
Out[8]: (array([ 1880.,  5019.,  5242.,  4896.,  4076.,  2613.,  4159.,
    1968.,  1734.,  3422., 10196.,  6642., 10257., 16832.,
    10814.,  8675.,  5720.,  2644.,  2242.,  970.]),
        array([ 9721. ,  9744.6,  9768.2,  9791.8,  9815.4,  9839. ,
    9862.6,  9886.2,  9909.8,  9933.4,  9957. ,  9980.6,
    10004.2, 10027.8, 10051.4, 10075. , 10098.6, 10122.2,
    10145.8, 10169.4, 10193. ]),
        <a list of 20 Patch objects>)
```



```
In [9]: print "El numero promedio de proteinas es de ",average(p2)," y su desviacion e
        standar es",std(p2)
        print "El numero promedio de RNAs es de ",average(r2)," y su desviacion estandar
        es",std(r2)
        print "El numero promedio entre eventos es de ",average(dt2)," y su desviacion
        estandar es",std(dt2)
```

El numero promedio de proteinas es de 9976.4213689 y su desviacion estandar es 114.768864809

El numero promedio de RNAs es de 991.580049272 y su desviacion estandar es 25.4741002957

El numero promedio entre eventos es de 0.0566920653984 y su desviacion estandar es 0.0567205523957

Comparación del primer y segundo caso

A primer vistazo, la desviación estándar en el segundo caso es bastante mayor, pero eso es porque el valor medio es también mucho mayor. Para tener una idea del ruido, queremos ver la desviación estándar relativa o coeficiente de variación, esto es, la desviación estándar sobre el valor medio, y comparamos esos valores. Así, para el primer caso y segundo caso respectivamente, obtenemos

```
In [16]: print "El coeficiente de variacion porcentual para los RNAs en el primer caso
es ",100*std(r)/mean(r)
print "El coeficiente de variacion porcentual para las proteinas en el primer
caso es ",100*std(p)/mean(p)
print "El coeficiente de variacion porcentual para los RNAs en el segundo caso
es ",100*std(r2)/mean(r2)
print "El coeficiente de variacion porcentual para las proteinas en el segundo
caso es ",100*std(p2)/mean(p2)
```

```
El coeficiente de variacion porcentual para los RNAs en el primer caso es 29.0
680921999
El coeficiente de variacion porcentual para las proteinas en el primer caso es
8.02456570628
El coeficiente de variacion porcentual para los RNAs en el segundo caso es 2.5
6904123015
El coeficiente de variacion porcentual para las proteinas en el segundo caso es
1.15040113649
```

Punto c

Como se puede ver, la variación porcentual es mucho menor para ambas distribuciones en el segundo caso, lo que hace que el ruido disminuya considerablemente. Sin embargo las células rara vez adquieren esta estrategia en la naturaleza, pues el costo metabólico es muy alto y en realidad no se justifica para la mayoría de los casos. La célula en muchos de sus procesos puede permitirse niveles de ruido relativamente altos.

In []: