

La solución a este taller debe subirse por SICUA antes de las 8:00AM del jueves 1 de septiembre del 2016.

(10 pt) Los archivos del código deben subirse en un único archivo **.zip** con el nombre **NombreApellido\_hw1.zip**, por ejemplo yo debería subir el zip **JaimeForero\_hw1.zip** al descomprimir el zip debe crearse la carpeta **JaimeForero\_hw1** y adentro debe estar el código.

## 1. Modelos genéticos

El ADN almacena la información genética de los seres vivos. Para expresar dicha información se dan los procesos de transcripción (en el cual se produce ARNm) y traducción (en el cual se producen proteínas).

Denotemos por  $r(t)$  y  $p(t)$  los números de ARNm y proteínas, respectivamente. Un modelo sencillo de expresión genética considera 4 eventos que pueden ocurrir: producción de ARNm con una tasa  $k_r$ , producción de proteína con una tasa  $k_p r(t)$ , degradación del ARNm con una tasa  $\gamma_r r(t)$  y degradación de proteína con tasa  $\gamma_p p(t)$ . La figura 1 representa dichos eventos.

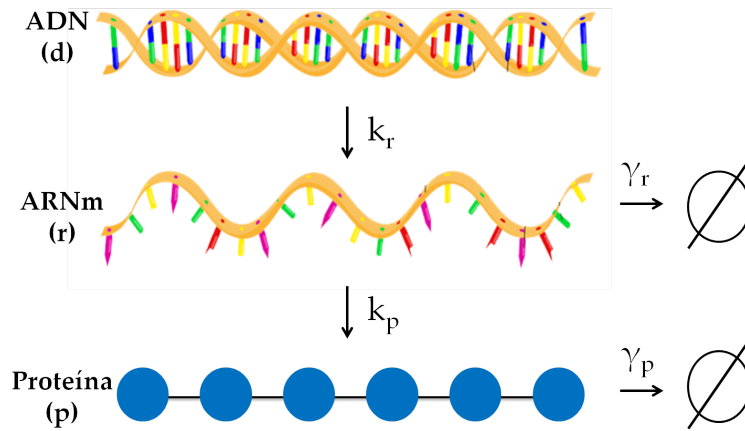


Figura 1: Procesos involucrados en la expresión genética.

Teniendo en cuenta lo anterior, las ecuaciones diferenciales que satisfacen  $r(t)$  y  $p(t)$  son

$$\begin{aligned} \frac{d}{dt} r(t) &= k_r - \gamma_r r(t), \\ \frac{d}{dt} p(t) &= k_p r(t) - \gamma_p p(t). \end{aligned} \tag{1}$$

Estas ecuaciones tienen una solución analítica:

$$\begin{aligned} r(t) &= \frac{k_r}{\gamma_r} (1 - e^{-\gamma_r t}), \\ p(t) &= \frac{k_r k_p}{\gamma_r \gamma_p} \left[ 1 - e^{-\gamma_p t} + \frac{\gamma_p}{\gamma_p + \gamma_r} (e^{-\gamma_p t} + e^{-\gamma_r t}) \right]. \end{aligned} \tag{2}$$

Sin embargo, ahora nos interesa considerar el aspecto discreto (es decir, que el número de ARNm y proteínas solamente puede tomar valores enteros) y aleatorio de este problema.

Para resolver las ecuaciones numéricamente teniendo en cuenta el carácter estocástico de los eventos se considera un pequeño intervalo  $\Delta t$ , en el cuál se evaluará si ocurre alguno de los eventos. La probabilidad de que ocurra algún evento en un tiempo  $\Delta t$  pequeño es

$$P(\text{evento}) = \text{Tasa del evento} \cdot \Delta t$$

Por ejemplo, la probabilidad para que en un intervalo  $\Delta t$  se produzca una proteína es  $k_p \cdot r(t) \cdot \Delta t$ , y análogamente para los demás eventos.

Para resolver las ecuaciones se deben generar números aleatorios en cada intervalo  $\Delta t$  para evaluar si cada uno de los eventos ocurre (de manera similar a como hicimos para la marcha aleatoria). Según los resultados se actualizan las cantidades de ARNm y proteínas. Esto se repite hasta llegar a un tiempo final  $T_f$ .

- (a) 10 pt Escriba un archivo llamado `genetica.py` que contenga la definición de la clase **Expresion** que puede inicializarse con los parámetros  $k_r, k_p, \gamma_r, \gamma_p, r_0 \equiv r(t=0), p_0 \equiv p(t=0), \Delta t$  y  $T_f$ . Los valores por defecto deben ser  $k_r = 1 \frac{\text{ARN}}{\text{min}}, k_p = 60 \frac{\text{proteinas}}{\text{ARN} \cdot \text{min}}, \gamma_r = \frac{1}{5} \frac{1}{\text{min}}$  y  $\gamma_p = \frac{1}{30} \frac{1}{\text{min}}, r_0 = 0, p_0 = 0, \Delta t = 0,0003$  minutos y  $T_f = 150$  minutos.
- (b) 20 pt Escriba una función llamada `resuelve` dentro de la clase **Expresion** para solucionar las ecuaciones diferencial de manera estocástica para cualquier valor de los parámetros iniciales. La función debe generar las listas de tiempo, número de ARN y número de proteínas. Estas listas deben ser parte de la clase **Expresion**.
- (c) 20 pt Escriba una función llamada `grafica` dentro de la clase **Modelo** que realice gráficas de  $r(t)$  vs.  $t$  y  $p(t)$  vs.  $t$  que deben ser guardadas en archivos de formato pdf llamados `r_t.pdf` y `p_t.pdf`. La función debe tomar como opción un argumento booleano llamado `analitica` para elegir si se grafican o no las ecuaciones (2) sobre la solución numérica.
- (d) 40 pt Escriba un código en python llamado `poblacion.py` que utilice a la clase **Expresion** **solamente con las funciones requeridas en los puntos anteriores** para
  - (20 pt) resolver las ecuaciones  $N = 200$  veces con los mismos parámetros iniciales y guardar en archivos de texto (`r_poblacion.dat`, `p_poblacion.dat`) el valor final de  $r$  y  $p$  para cada ejecución.
  - (20 pt) hacer histogramas (por separado, guardados en archivos `r_histograma.pdf` y `p_histograma.pdf`) para el número de ARN y uno para el número de proteínas. El histograma para el número de ARN debe incluir una comparación con el resultado analítico esperado que es una distribución de Poisson dada por la ecuación.

$$P(r) = \frac{\lambda^r e^{-\lambda}}{r!},$$

con  $\lambda = \frac{k_r}{\gamma_r}$ . Para utilizar el factorial ejecute la función  $\Gamma$  de tal forma que  $\Gamma(r+1) = r!$ . Esto se puede hacer de la siguiente manera:

```
import scipy.special as ssp
ssp.gamma(r)
```

**NOTA:** Este código debe funcionar para cualquier valor de los parámetros de inicialización de la clase **Expresion**.