

Introducción: repeticiones

Curso 0



DIPLOMA EXPERTO

Bioinformática
y Genómica Computacional



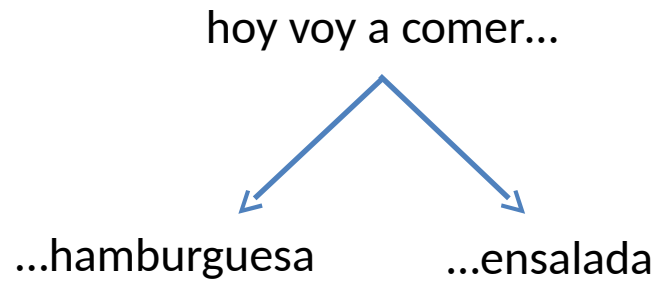
Rodrigo Santamaría
2020



Control de flujo

Control de flujo

Elegir



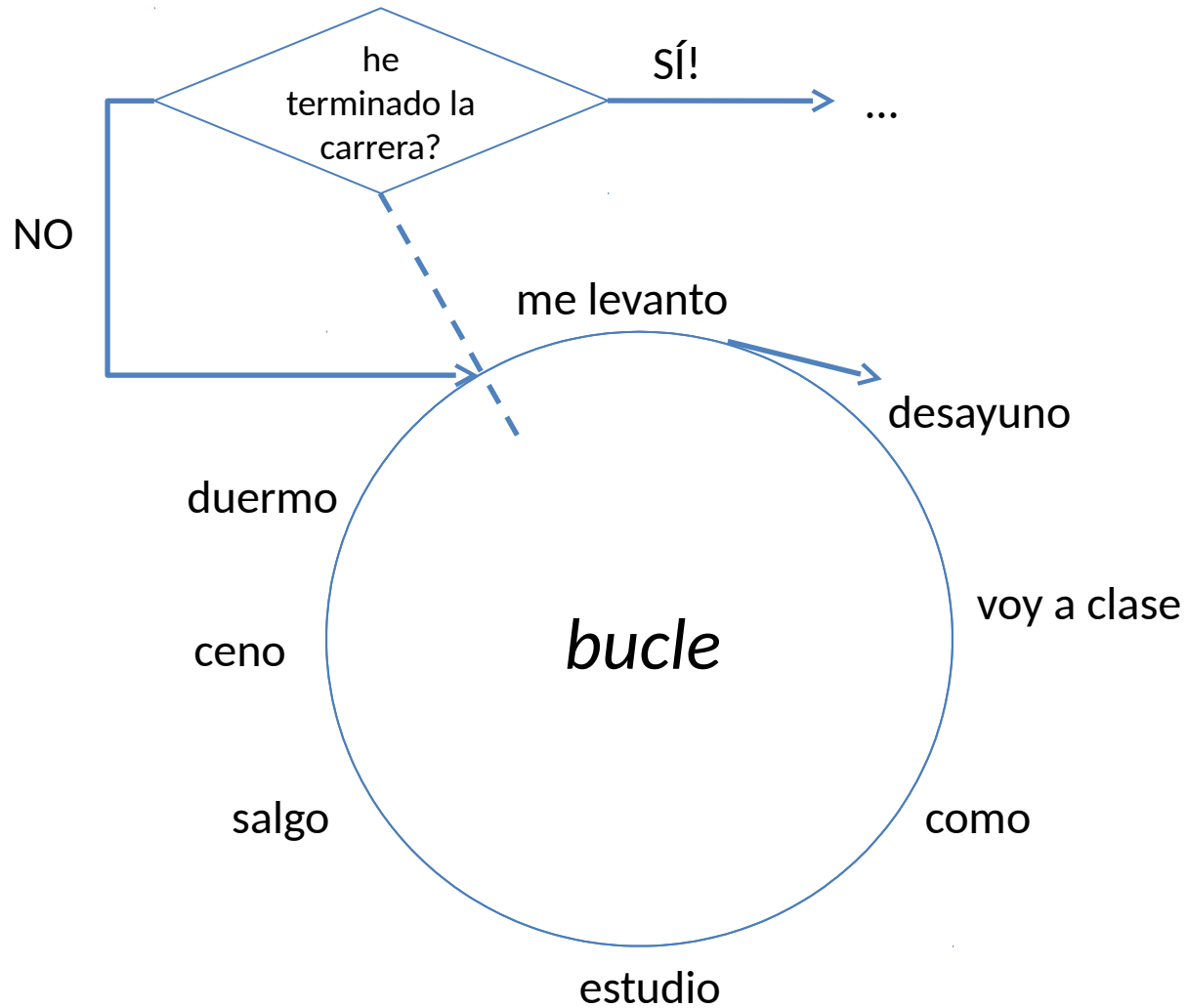
Repetir





Repeticiones

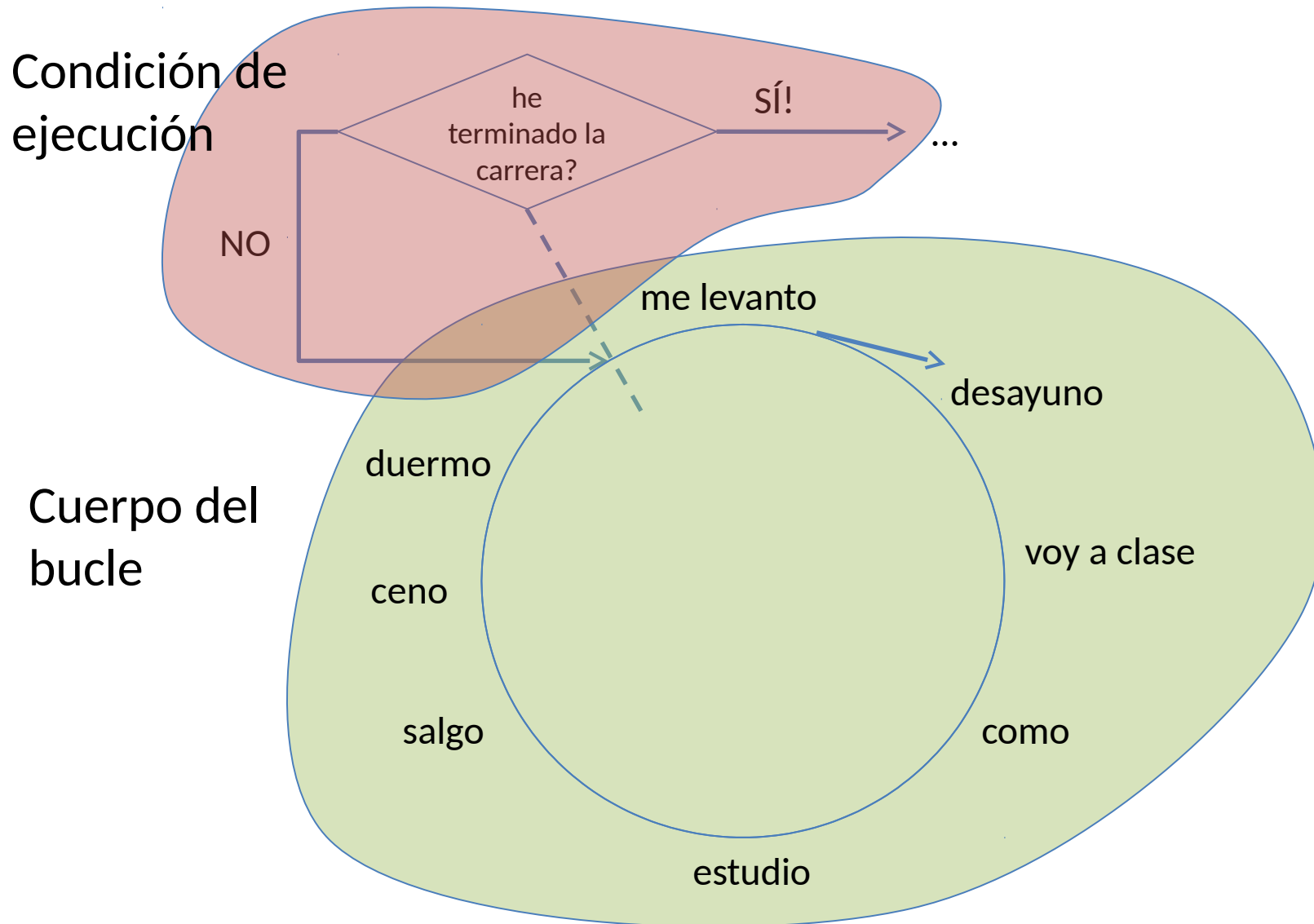
Repetición



En inglés se llama *loop*, que podríamos traducir como bucle o tirabuzón

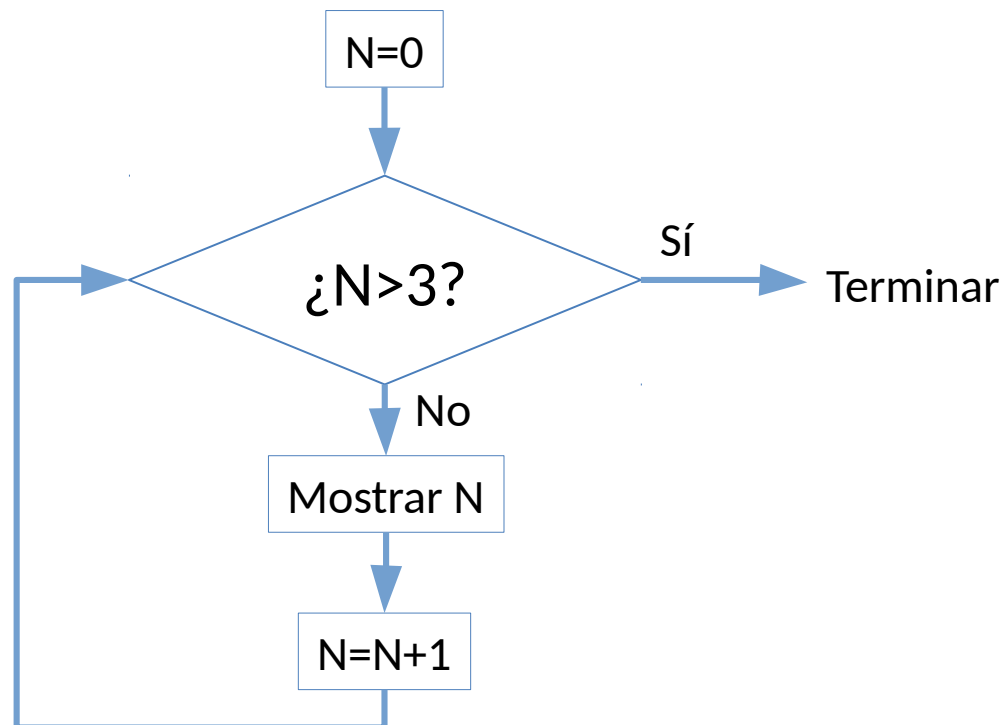


Repetición



Iteración

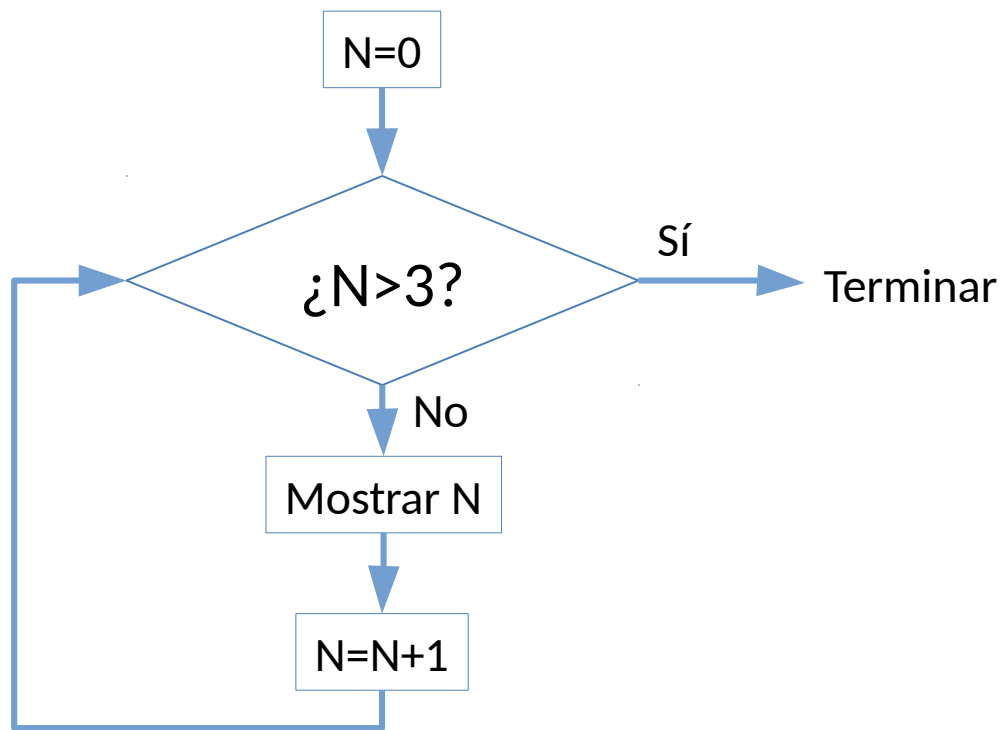
- Cada una de las ejecuciones de todas las sentencias del cuerpo del bucle



¿De cuántas iteraciones consta este bucle?

‘Desenrollar’ un bucle

- Describir lo que ocurre en *todas* sus iteraciones



N=0		
0>3	Falso	
Mostrar 0		1ª iteración
N=0+1		
1>3	Falso	
Mostrar 1		2ª iteración
N=1+1		
2>3	Falso	
Mostrar 2		3ª iteración
N=2+1		
3>3	Falso	
Mostrar 3		4ª iteración
N=3+1		
4>3	Verdadero	
Terminar		

Bucle con condición

- 1) se evalúa la condición
- 2) **si** la condición es verdadera
se ejecuta el cuerpo
se vuelve al paso 1
si no se abandona el bucle

```
mientras condición  
    hacer  
        sentencia  
fin_mientras
```

Sentencia WHILE

while (*condición*):
 sentencia_1
 ...
 sentencia_n



Condición del bucle: expresión que se evalúa en cada vuelta (iteración) del bucle.

Si es cierta, se ejecuta el cuerpo, si no, sale del bucle

Cuerpo del bucle: conjunto de sentencias que se van a repetir mientras *condición* sea cierta

```
n=0
while n<10:
    print("El cuadrado de", n, "es ", n**2)
    n=n+1
```

```
El cuadrado de 0 es 0
El cuadrado de 1 es 1
El cuadrado de 2 es 4
El cuadrado de 3 es 9
El cuadrado de 4 es 16
El cuadrado de 5 es 25
El cuadrado de 6 es 36
El cuadrado de 7 es 49
El cuadrado de 8 es 64
El cuadrado de 9 es 81
```

Bucle sobre un conjunto

1. Se elige un conjunto
2. Se elige el siguiente elemento del conjunto
Se ejecuta el cuerpo del bucle
3. Si no quedan elementos en el conjunto, termina

```
para cada elemento en conjunto  
  hacer  
    sentencia  
fin_para_cada
```

Sentencia FOR

```
for x in colección:  
    sentencia_1  
    ...  
    sentencia_n
```

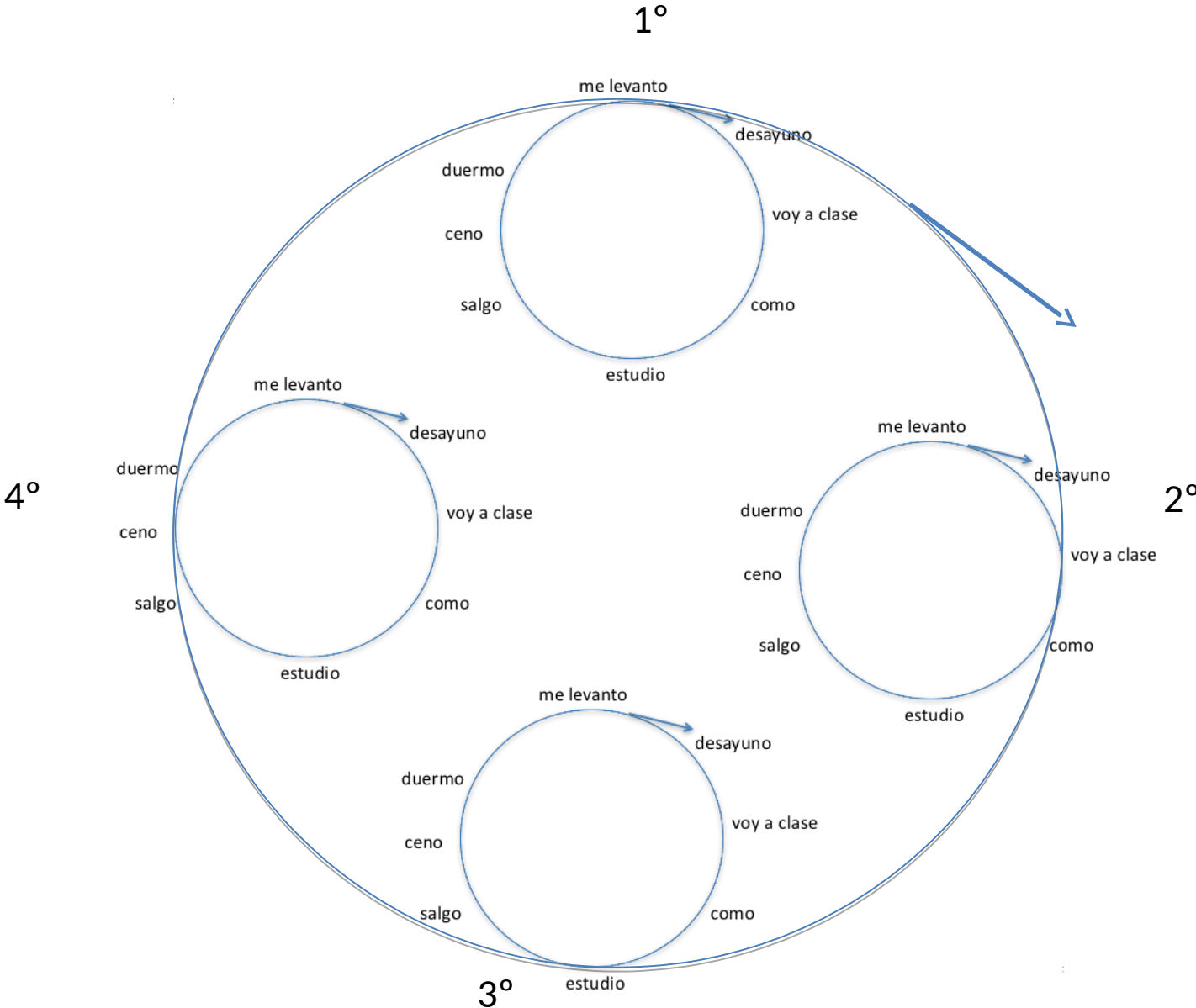
→ Para cada elemento x en un array*

Cuerpo del bucle: conjunto de sentencias que se van a repetir en cada iteración del bucle, cada vez con x conteniendo un elemento de la lista

```
for n in [1, 2, 3, 4, 5]:  
    print "El cuadrado de ", n, " es ", n*n
```

* Vimos en la segunda sesión que un array puede ser una cadena de texto, o un vector de números. En general, un vector en python se representa como números separados por comas, entre corchetes

Bucles anidados



Bucles anidados

```
para cada año en [1,2,3,4]
  para cada dia en [1, ..., 365]
    Me levanto
    Desayuno
    Voy a clase
    Como
    Estudio
    Salgo
    Ceno
    Duermo
  fin para cada
fin para cada
```

En el cuerpo del mensaje,
las variables año y dia irán
tomando valores distintos
con cada iteración

Bucle infinito

- Es un bucle donde la condición de ejecución nunca es falsa
 - Generalmente, son producto de errores en el código
- Sentencia **break**
 - Termina con el bucle

```
n=0  
while True:  
    n=n+1
```

Bucle infinito

```
n=0  
while True:  
    n=n+1  
    if n==1000:  
        break
```

Bucle infinito que se rompe cuando n vale 1000

Ejercicio 1

- Desenrolla el siguiente bucle:

```
i=8
```

```
h=0
```

```
mientras i>=0:
```

```
    h=i-3
```

```
    i=i-h
```


Ejercicio 2

- Tenemos una lista con los siguientes niveles de expresión para varios genes:

`exp = [4.5, 8.3, 7.7, 9.0]`

- Para normalizarlos por la cobertura de este experimento, debemos dividirlos entre 3.1
- Realiza un programa que, mediante un bucle, los normalice y almacene en la lista `expn`*

```
* #definimos una lista vacía así:  
expn=[]  
#para añadir elementos a una lista:  
expn.append(3)  
expn.append(67)  
expn
```

[3, 67]

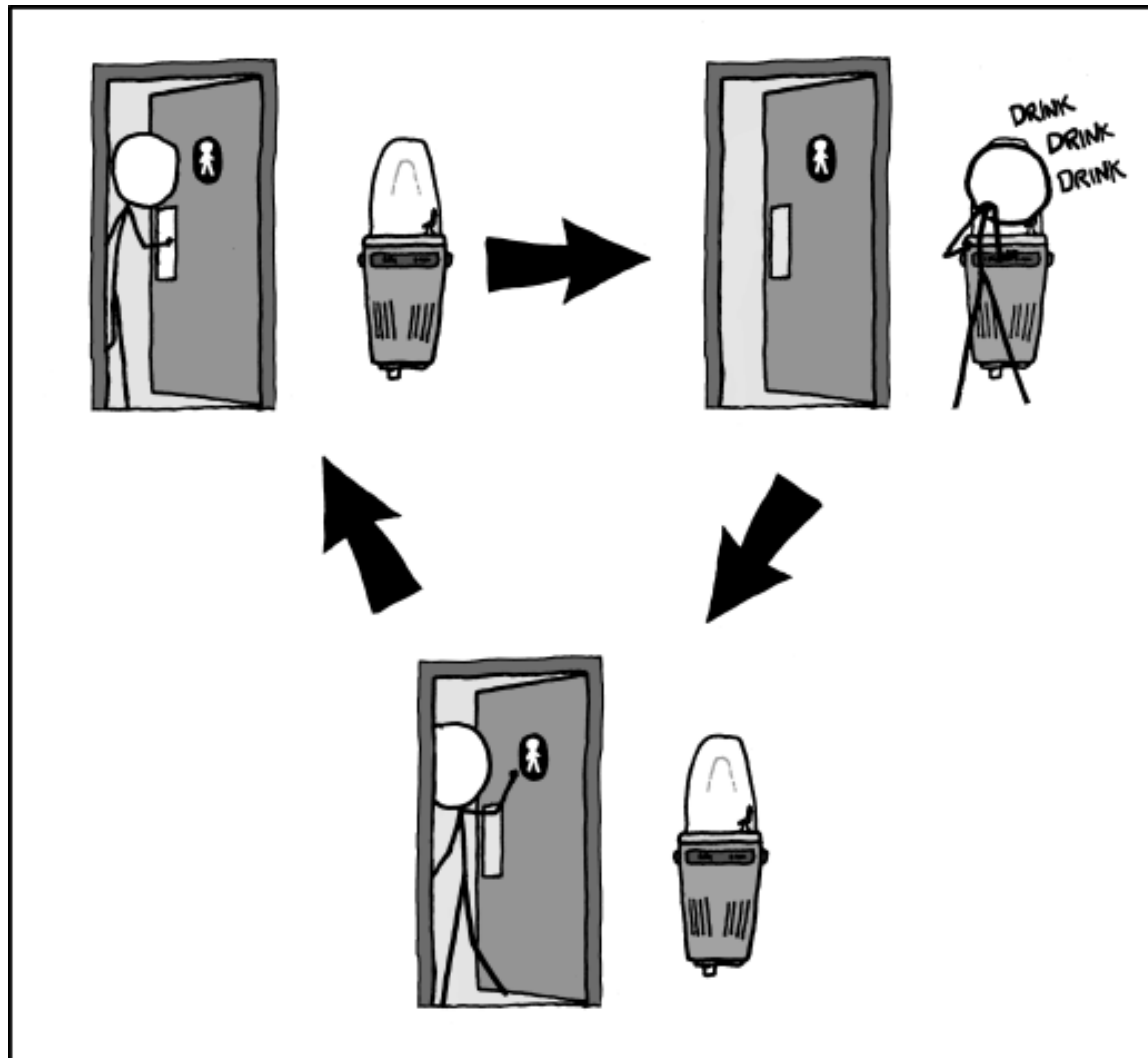
Ejercicio 3

- Tenemos la secuencia de nucleótidos de SCNN1A
 - Guárdala en una variable `scnn1a`
- Calcular su cadena complementaria y almacénala en `scnn1a_comp`
 - Sin cambiar el sentido, por ejemplo:
 - AACCTCAG → TTGGAGTC

```
* #también podemos definir cadenas vacías:  
scnn1a_comp=""  
#Y añadir letras con el operador +  
scnn1a_comp=scnn1a_comp+"A"  
scnn1a_comp=scnn1a_comp+"TT"  
scnn1a_comp
```

'ATT'

Repetición



<http://xkcd.com/986>

I AVOID DRINKING FOUNTAINS OUTSIDE BATHROOMS
BECAUSE I'M AFRAID OF GETTING TRAPPED IN A LOOP.