

In []:

INGENIERÍA EN BIOTECNOLOGÍA

GBI6 - BIOINFORMÁTICA



Estudiante: Juleth Belsabeth Flores Guerrero

Edad: 21

Domicilio: Napo-Baeza

Nacionalidad: Ecuatoriana

Correo: juleth.flores@est.ikiam.edu.ec

Fecha: 17/07/2022

Se produzcan nuevas generaciones a partir de la formación aleatoria de alelos provenientes

In []:

ACTIVIDAD

Crear un módulo alelos.py con las tres funciones de "Creación de la población", "Cuantificación de la población" y "Selección natural".
Crear un Jupyter Notebook tarea6_alelos.ipynb, cargar el módulo alelos y ejecutar las funciones.
Crear un repositorio en GitHub de nombre tarea6 y guardar los dos archivos de la tarea. El enlace del repositorio en la actividad T6_alelos de Moodle

In []:

```
import alelos
```

In []:

```
alelos.build_population(25, 0.43)
```

In []:

```
[('A', 'a'),  
 ('a', 'a'),  
 ('a', 'A'),  
 ('A', 'a'),  
 ('a', 'a'),  
 ('a', 'a'),  
 ('A', 'a'),  
 ('a', 'a'),  
 ('A', 'A'),  
 ('A', 'a'),  
 ('a', 'a'),  
 ('a', 'a'),  
 ('a', 'a'),  
 ('a', 'A'),  
 ('A', 'A'),  
 ('A', 'a'),  
 ('a', 'A'),  
 ('a', 'A'),  
 ('a', 'a'),  
 ('A', 'a'),  
 ('A', 'A'),  
 ('A', 'a'),  
 ('A', 'a'),  
 ('A', 'a'),  
 ('a', 'a')]
```

In []:

```
nuevo=alelos.build_population(10, 0.35)  
print(nuevo)  
  
[( 'a', 'A'), ('A', 'a'), ('a', 'a'), ('a', 'a'), ('a', 'a'), ('a', 'A'), ('a', 'a'), ('a',
```

In []:

```
conteoalelo = alelos.compute_frequencies(nuevo)  
print(conteoalelo)  
  
{ 'AA': 0, 'aa': 6, 'Aa': 2, 'aA': 2 }
```

In []:

```
crecimiento = alelos.reproduce_population(nuevo)  
print(crecimiento)  
  
[( 'A', 'a'), ('a', 'a'), ('a', 'a'), ('a', 'a'), ('a', 'a'), ('a', 'a'), ('a', 'a'), ('A',
```

In []:

```
conteoalelo2 = alelos.compute_frequencies(crecimiento)  
print(conteoalelo2)  
  
{ 'AA': 0, 'aa': 8, 'Aa': 2, 'aA': 0 }
```

In []:

```
def simulate_drift(N, p):
    # initialize the population
    my_pop = alelos.build_population(N, p)
    fixation = False # condición inicial de fijación
    num_generations = 0 # población parental
    while fixation == False:
        # compute genotype counts
        genotype_counts = alelos.compute_frequencies(my_pop)
        # if one allele went to fixation, end
        if (genotype_counts["AA"] == N or genotype_counts["aa"] == N):
            print("An allele reached fixation at generation", num_generations)
            print("The genotype counts are")
            print(genotype_counts)
            fixation == True
            break
        # if not, reproduce
        my_pop = alelos.reproduce_population(my_pop)
        num_generations += 1
    return num_generations, genotype_counts
```

In []:

```
sim1 = simulate_drift(100, 0.5 )
sim1
```

In []:

```
An allele reached fixation at generation 254
The genotype counts are
{'AA': 0, 'aa': 100, 'Aa': 0, 'aA': 0}
(254, {'AA': 0, 'aa': 100, 'Aa': 0, 'aA': 0})
```

In []:

```
Generacion100= alelos.build_population(100,0.5) #resultado 100 individuos con una probabili
print(Generacion100) # observar la generacion de 100
Conteo= alelos.compute_frequencies(Generacion100) # conteo del numero de alelos en generaci
print(Conteo) #se imprimio este conteo
len(Generacion100) # conocer el numero de individuos en esta generacion
```

In []:

```
[('A', 'a'), ('A', 'A'), ('A', 'A'), ('a', 'a'), ('a', 'A'), ('a', 'a'), ('a', 'A'), ('a',
{'AA': 17, 'aa': 23, 'Aa': 33, 'aA': 27}
100
```