Journées de l'APMEP 2021 - 2022

Programmation en Python des mécanismes de l'évolution

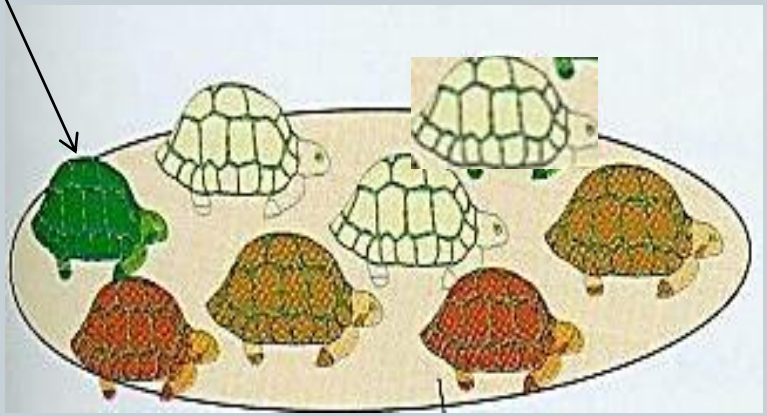
Anne Juras et Jean-Louis MARCIA
Groupe de travail MATH-SVT

Sommaire

- Le concept de l'évolution en SVT
- Un peu de génétique : allèles, génotypes et reproduction
- La loi de Hardy-Weinberg
- Simuler l'évolution des fréquences alléliques au cours des générations
    - Modélisation
    - Les fonctions python et le corps du programme
    - Premiers résultats
    - Analyse des résultats
- Comment intégrer les autres forces évolutives ?
    - Mutations
    - Sélection naturelle
    - Appariement non aléatoire
- Ressources

Le concept d'évolution en SVT

Mutation



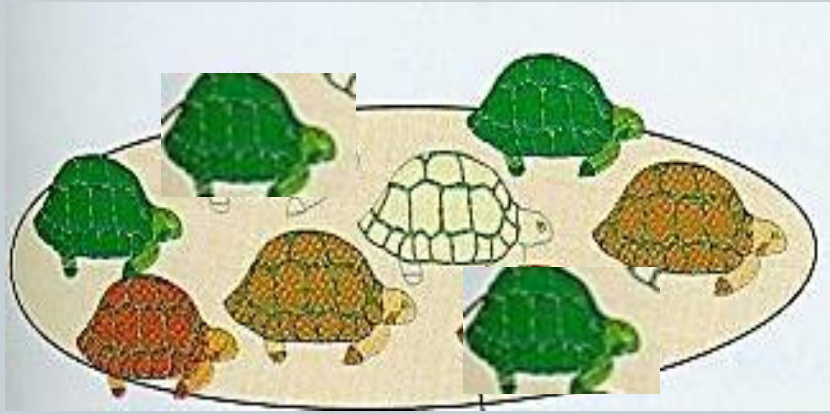
Espèce A
Population génération 0

Plusieurs générations
= au cours du temps

▪Définition d'un ***paramètre*** observable/mesurable dont la variation rend compte d'un changement/d'une évolution **FREQUENCE ALLELIQUE** (support génétique)

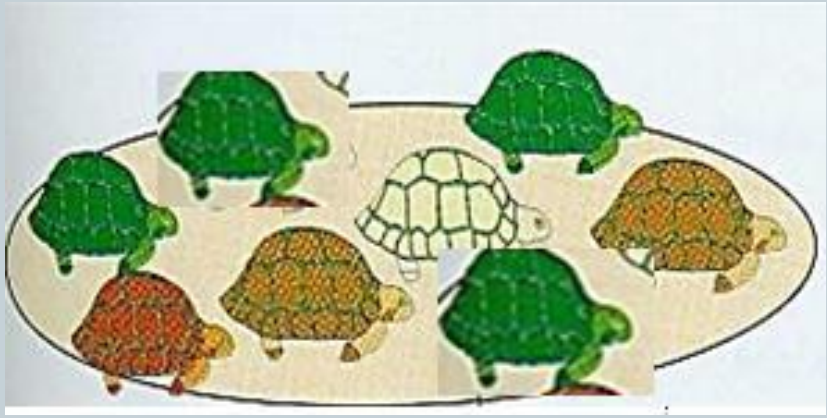
•Définition de « forces évolutives » capable de faire varier le paramètre:

- •Mutation
- •Dérive génétique
- •Sélection naturelle



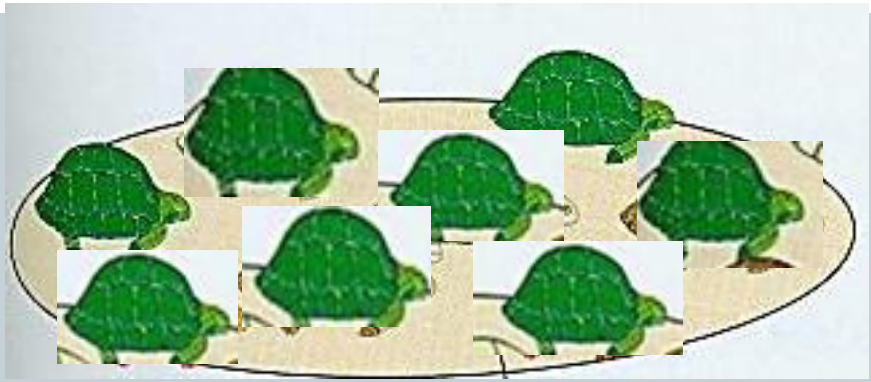
Espèce A
Population génération 100

Le concept d'évolution en SVT



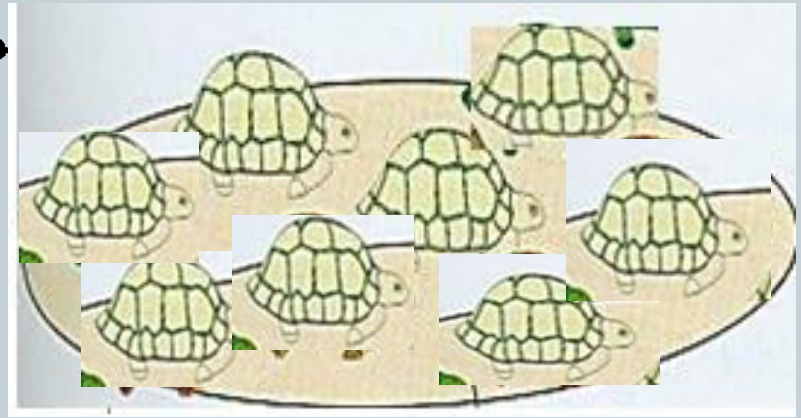
Espèce A
Population génération 100

plusieurs générations
= au cours du temps

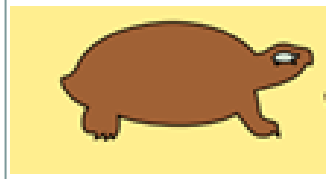


Espèce A'

SPECIATION



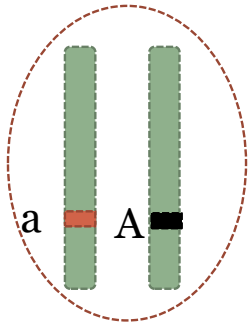
Espèce A"



Reproduction des individus

Extrait du PNF enseignement scientifique de terminale – Thème 3.1

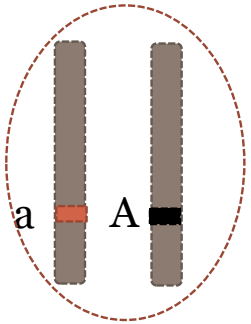
Génération g



Individu 1

Allèles : différentes versions d'un gène
Dans ce schéma: deux allèles A et a d'un même gène

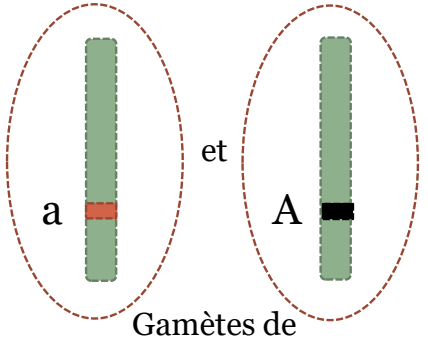
Génotype : les deux allèles présents chez un individu pour un gène donné



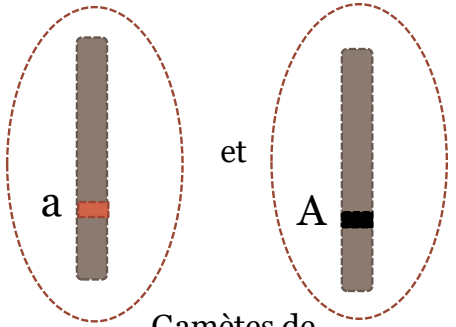
Individu 2

MÉIOSE

Séparation des chromosomes de la même paire



Gamètes de l'individu 1



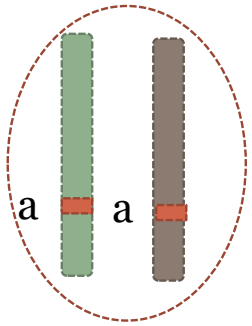
Gamètes de l'individu 2

FÉCONDATION : Rencontre **aléatoire** de deux gamètes

gamètes	A	a
A	A//A	A//a
a	A//a	a//a

Génotypes possibles pour la génération n+1 : A//a ou a//a ou A//A

Génération g+1

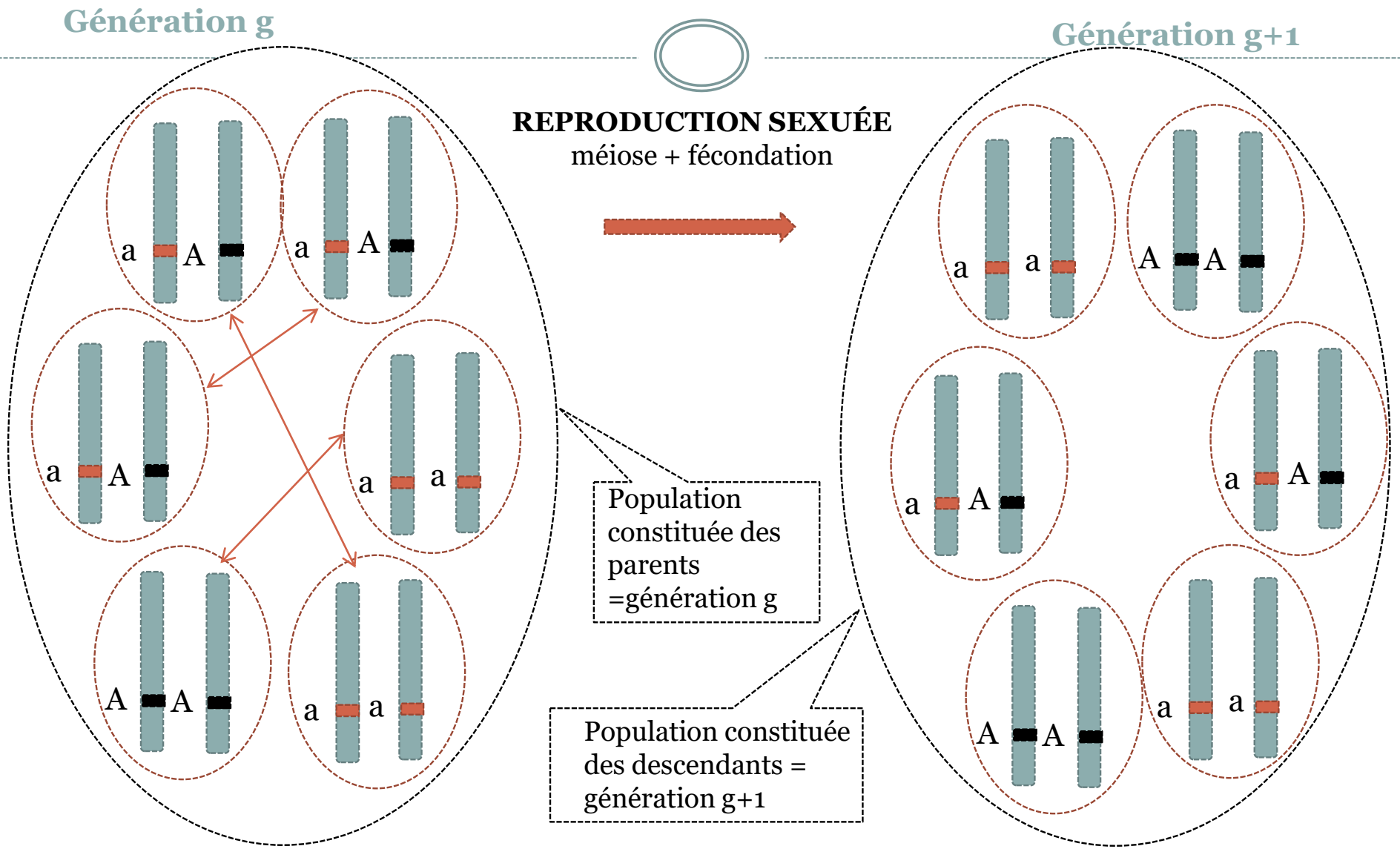


Descendant

Résultat

Génotype du descendant : a//a pour le gène considéré

À l'échelle d'une population



La loi de Hardy-Weinberg

Hypothèses :

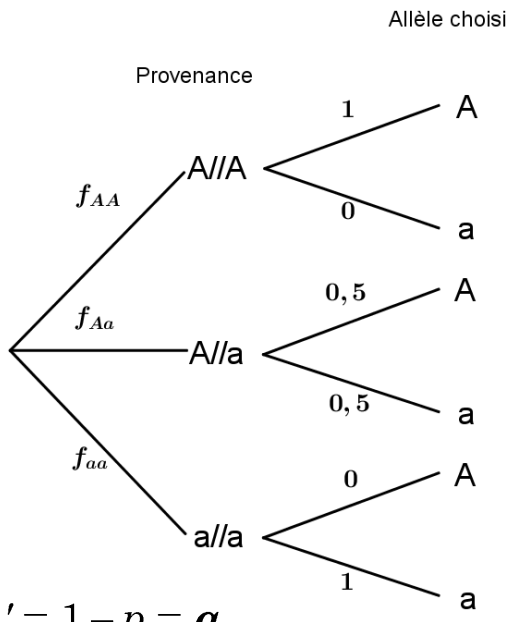
- La taille de la population est supposée infinie ce qui permet d'assimiler les probabilités à des fréquences (loi des grands nombres)
- Le choix du partenaire se fait au hasard
- Pas de migration, ni mutation, ni sélection naturelle
- Les générations sont séparées

Notations :

- p et q les fréquences respectives des allèles A et a
- f_{AA}, f_{Aa}, f_{aa} celles des génotypes respectifs A//A, A//a et a//a

Liens entre les fréquences alléliques et les fréquences génotypiques :

$$\boxed{p = f_{AA} + \frac{1}{2} f_{Aa}} \; ; \; \boxed{q = f_{aa} + \frac{1}{2} f_{Aa} = 1 - p}$$



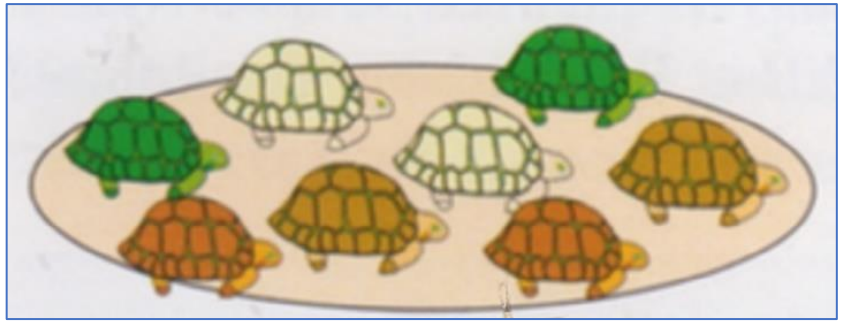
Vers la génération suivante : on note avec un « prime » les fréquences

$$\boxed{f'_{AA} = p^2} \; ; \; \boxed{f'_{Aa} = 2pq} \; ; \; \boxed{f'_{aa} = q^2}$$

Calcul de *p'* et *q'* : $\boldsymbol{p'} = f'_{AA} + \frac{1}{2} f'_{Aa} = p^2 + pq = p(p+q) = p \times 1 = \boldsymbol{p}$ et $\boldsymbol{q'} = 1 - p' = 1 - p = \boldsymbol{q}$.

Conclusion : Les fréquences alléliques sont stables à partir de la génération 0 et à partir de la génération 1 pour les fréquences génotypiques.

Que devient la loi de H.W. avec une population de taille finie ? Un exemple chez des tortues



On imagine une espèce de tortues dont la carapace a trois couleurs possibles (phénotypes) : marron, verte ou beige.

Le gène contrôlant la couleur de la carapace existe sous deux versions possibles : l'allèle **m** responsable de la couleur **marron** et l'allèle **v** responsable de la couleur **verte**.

Ces allèles sont co-dominants : le génotype **m//m** révèle la couleur **marron**

le génotype **m//v** révèle la couleur **beige**

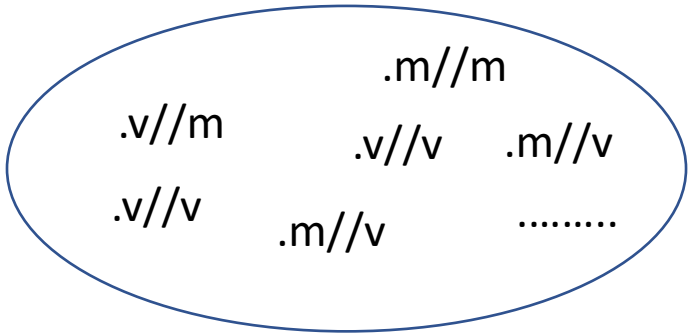
le génotype **v//v** révèle la couleur **verte**

Vers la simulation

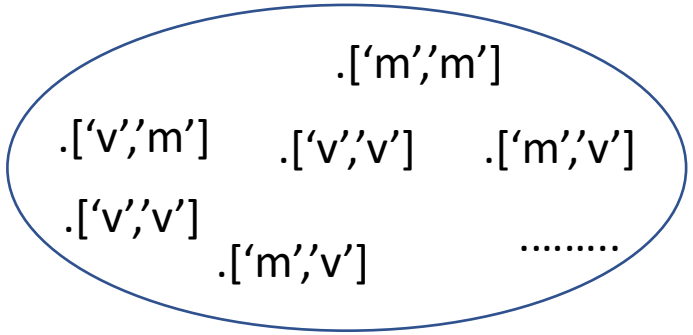
- Comment modéliser cette population de tortues ?
- Quelles hypothèses doit-on retenir ?
- Comment construire le programme renvoyant l'évolution des fréquences alléliques au cours des générations ?

Hypothèses de travail

- Chaque individu est assimilé à son génotype.
- Dans un premier temps, on suppose qu'il n'y a pas de mutation au cours des générations, ni de sélection naturelle et que les appariements se font aléatoirement.
- On considère que la fréquence des allèles dans la population ne dépend pas du sexe des individus. Les couples seront donc formés par le choix aléatoire de deux individus de cette population.
- On travaille avec une population à effectif constant au cours des générations. Chaque couple engendre donc deux descendants sans chevauchement entre génération.



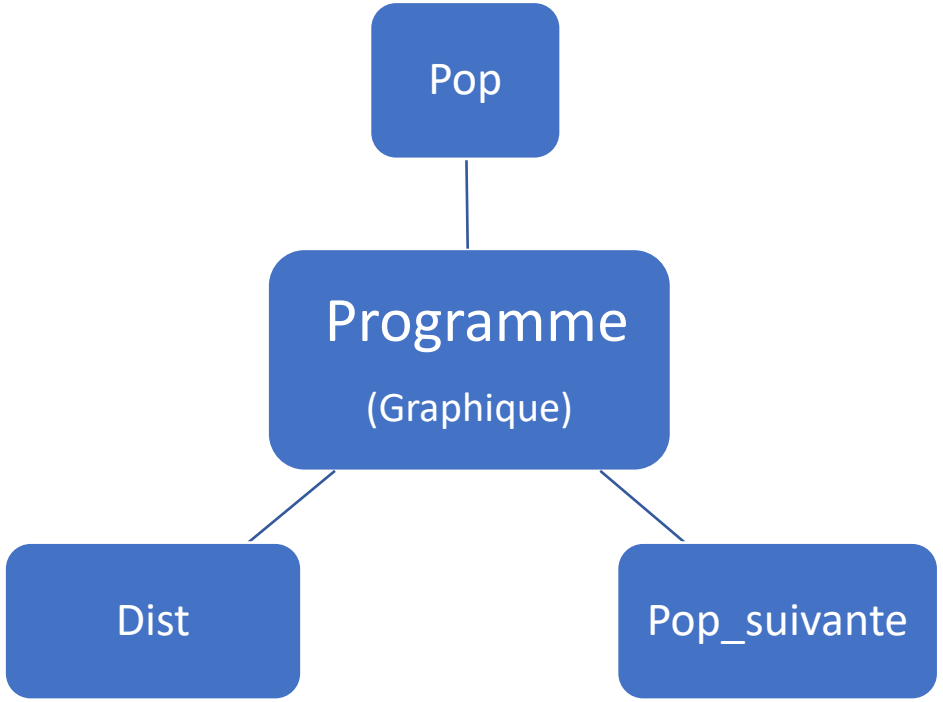
Population de génotypes



Traduction en python

Comment faire « évoluer » cette population ?

Les fonctions Python et le corps du programme :



Les fonctions

```
def Pop(n):
    U=['m','v']
    P=[]
    for i in range(2*n):
        I=[choice(U),choice(U)]
        P.append(I)
    return P
```

```
def Dist(P):
    N_m=0
    N_v=0
    for I in P:
        N_m=N_m+I.count('m')
        N_v=N_v+I.count('v')
    return [N_m/(2*len(P)),N_v/(2*len(P))]
```

```
def Pop_suivante(P):
    P_suivant=[]
    while P!=[]:
        I_1=choice(P)
        P.remove(I_1)
        I_2=choice(P)
        P.remove(I_2)
        E_1=[choice(I_1),choice(I_2)]
        E_2=[choice(I_1),choice(I_2)]
        P_suivant.append(E_1)
        P_suivant.append(E_2)
    return P_suivant
```

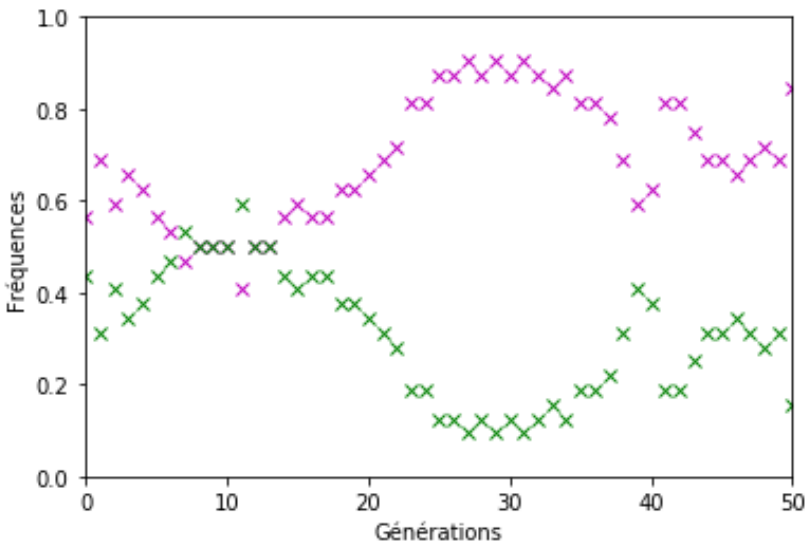
Le programme

```
import matplotlib.pyplot as plt

n = int(input("Nombre de couples dans la population : "))
g = int(input("Nombre de générations : "))

plt.axis([-0.1,g,0,1])
plt.xlabel('Générations')
plt.ylabel('Fréquences')

P = Pop(n)
L = Dist(P)
plt.plot(0,L[0],'m.')
plt.plot(0,L[1],'gx')

for i in range(1,g+1):
    P = Pop_suivante(P)
    L = Dist(P)
    plt.plot(i,L[0],'m.')
    plt.plot(i,L[1],'gx')

plt.show()
```

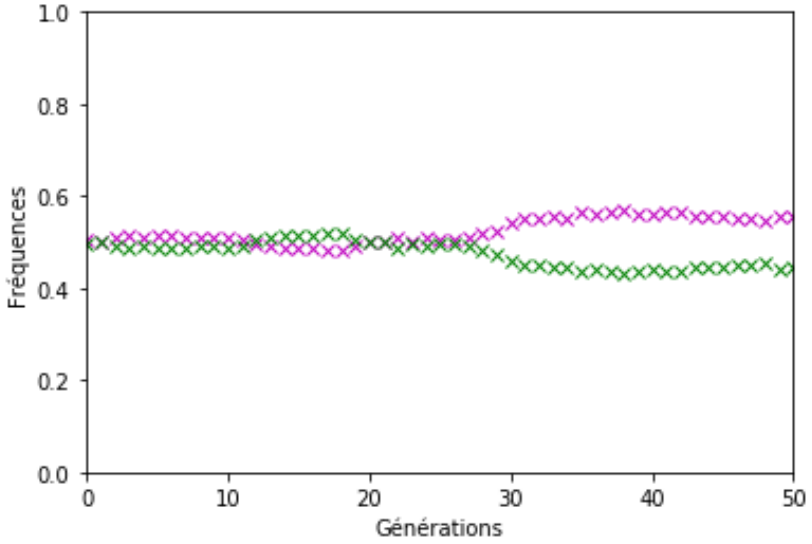
Vers ressources

Premiers résultats

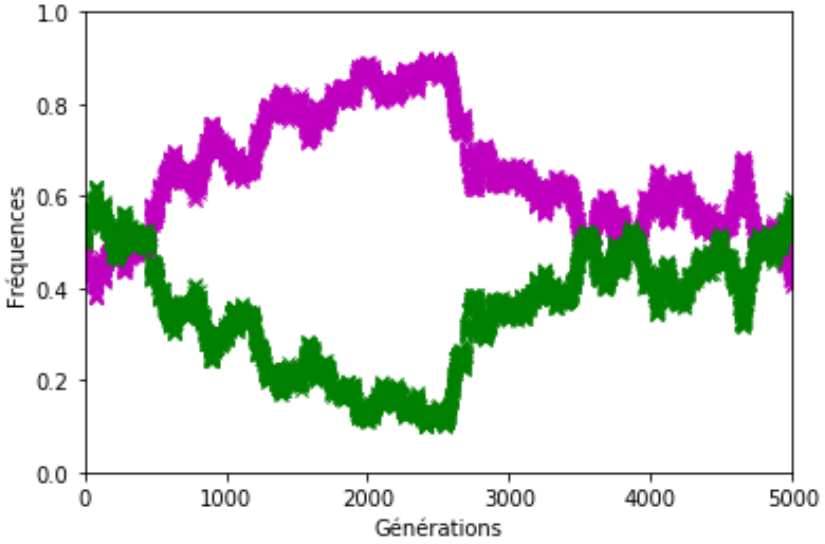
8 couples sur 50 générations



800 couples sur 50 générations



800 couples sur 5000 générations

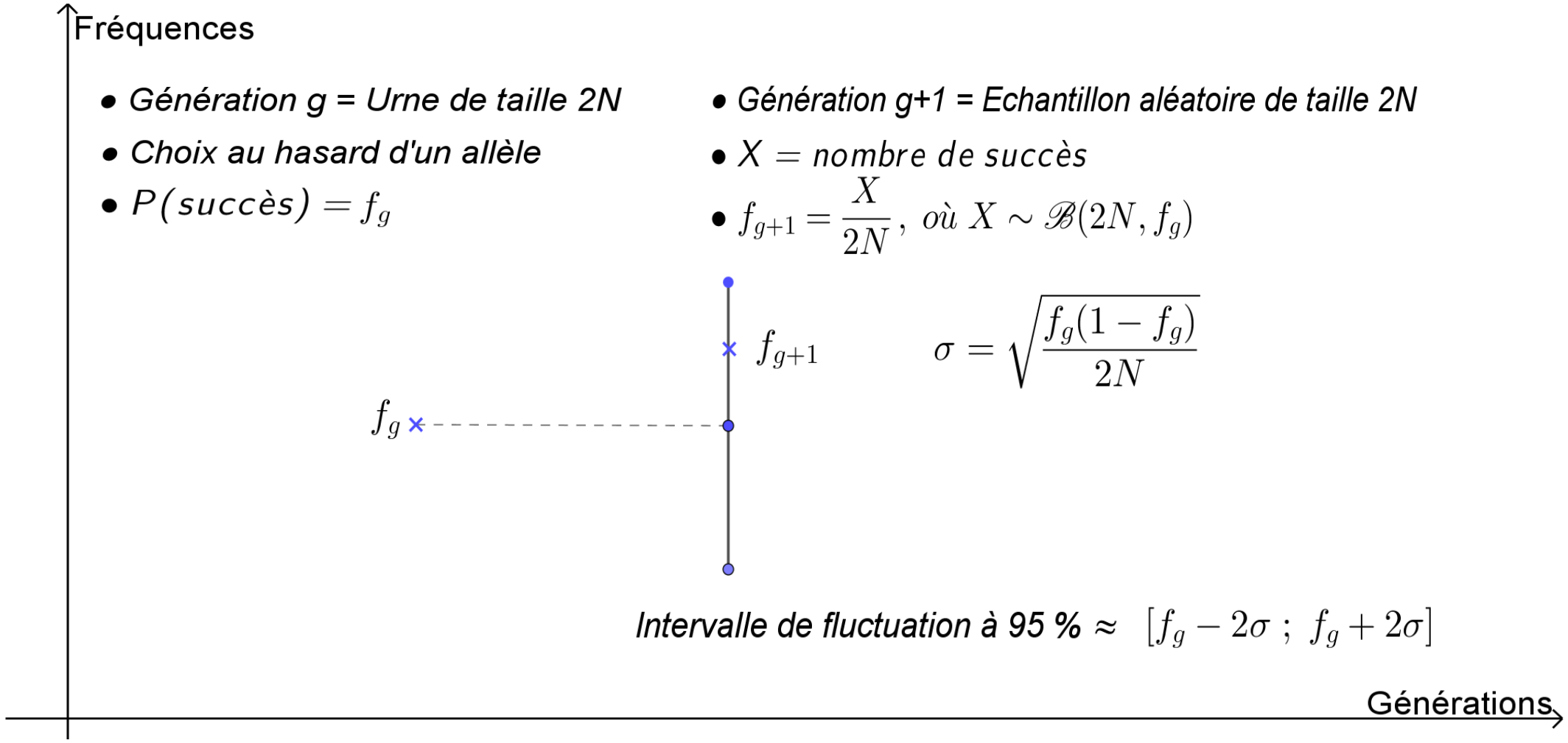


Cette fluctuation des fréquences alléliques au cours des générations est appelée **« dérive génétique ».**
Cela fait penser à une marche aléatoire.

Analyse des résultats - La dérive génétique

On note N la taille de la population (N = 2n)

Pour simplifier, on considère que la génération g+1 s'obtient par 2N répétitions indépendantes du choix au hasard d'un allèle dans la génération g.



Les autres forces évolutives

Mutations génétiques

En SVT: Une mutation est un changement de séquence d'un gène qui apparait aléatoirement et qui peut conduire à l'apparition d'un nouvel allèle chez un individu de la population

La fonction " mut "

La fonction "mut" s'applique à une population P formée de 2×n individus. Elle simule les mutations que peuvent subir les 2 allèles de chaque individu avec une fréquence f.
L'allèle muté (ou l'ensemble des allèles mutés) **est noté 'a'**.

```
def mut(P,f):
    for I in P:
        if random()<f:
            I[0]='a'
        if random()<f:
            I[1]='a'
    return P
```

- Comment intégrer cette fonction dans le programme ?
- Faut-il modifier les autres fonctions ?

Solution

Les fonctions Pop et Pop_suivante sont inchangées.
Seule la fonction Dist est modifiée :

```
def Dist(P):
    N_m=0
    N_v=0
    N_a=0
    for I in P:
        N_m=N_m+I.count('m')
        N_v=N_v+I.count('v')
        N_a=N_a+I.count('a')
    return [N_m/(2*len(P)),N_v/(2*len(P)),N_a/(2*len(P))]
```

- Que choisir pour f ?

Vers ressources

Intégration dans le programme :

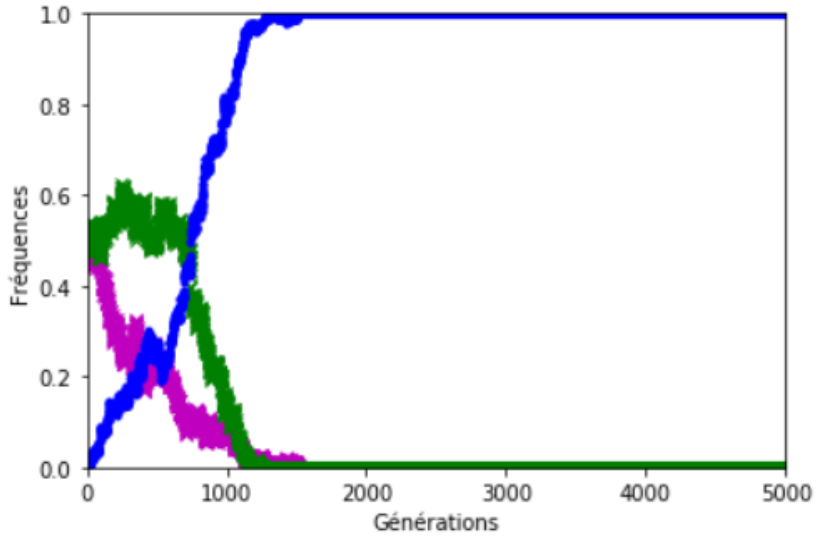
```
import matplotlib.pyplot as plt
from random import *

n=int(input("Nombre de couples dans la population : "))
g=int(input("Nombre de générations : "))
f=float(input("fréquence des mutations : "))

plt.axis([0,g,0,1])
plt.xlabel('Générations')
plt.ylabel('Fréquences')

P=Pop(n)
L=Dist(P)
M=[L[0]]
V=[L[1]]
A=[L[2]]

for i in range(g):
    P=Pop_suivante(P)
    P=mut(P,f)
    L=Dist(P)
    M.append(L[0])
    V.append(L[1])
    A.append(L[2])

abs=[k for k in range(g+1)]
plt.plot(abs,M,'mx')
plt.plot(abs,V,'gx')
plt.plot(abs,A,'b.')
plt.show()
```

Résultats mutations

Nombre de couples dans la population : 500

Nombre de générations : 5000

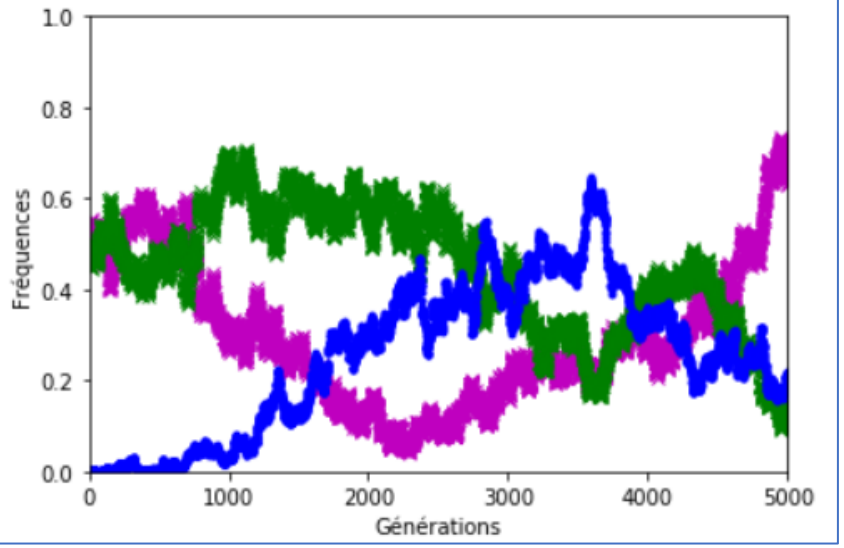
fréquence des mutations : 0.001



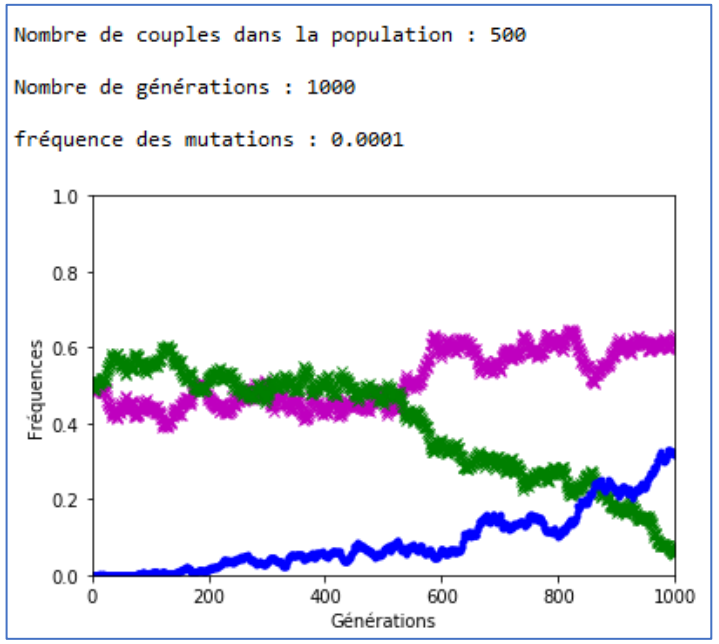
Nombre de couples dans la population : 500

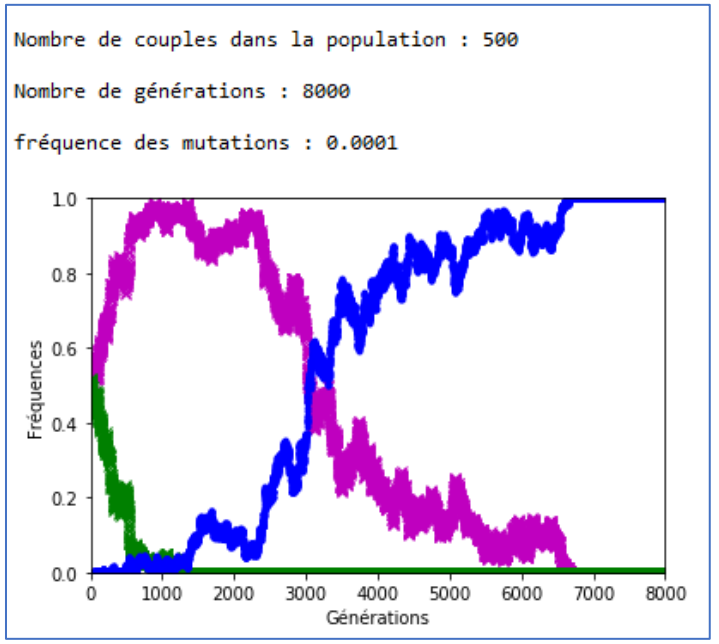
Nombre de générations : 5000

fréquence des mutations : 0.0001

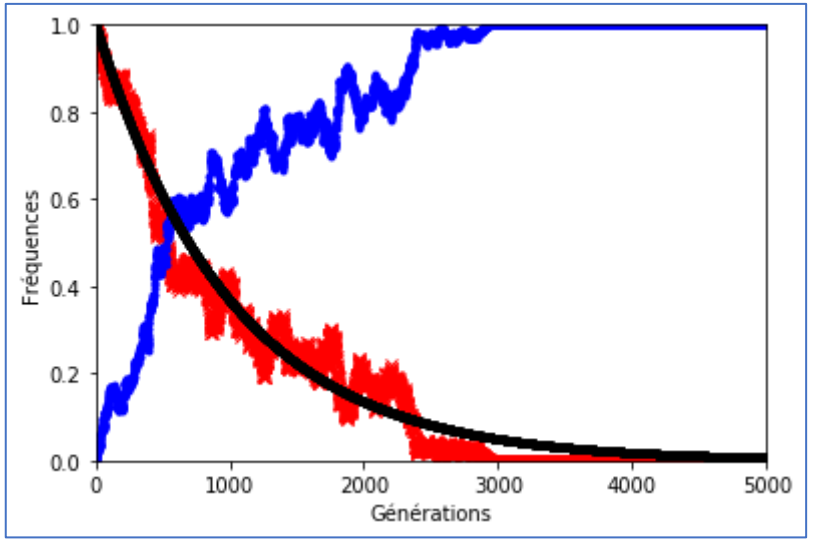


En attendant l'extinction des allèles d'origine





En l'absence de dérive génétique, les allèles d'origine seraient en **décroissance géométrique de raison 1–f** (représentée en **noir** sur le graphique).
Avec dérive génétique, la fréquence des allèles d'origine est représentée en **rouge** et celle des allèles mutés en **bleu**.



L'extinction G vérifie : $(1 - f)^G < 1/2N$.

Sélection naturelle

La fonction selec

La fonction "selec" s'applique à une population P formée de 2×n individus. Elle simule la sélection naturelle que peuvent subir certains génotypes. On choisit ici de simuler une sélection défavorable au phénotype associé au génotype ['m', 'm'] en le faisant disparaitre de la population avec une fréquence s.

Conséquence : L'effectif de la population n'est plus constant au cours des générations.

- Créer cette fonction sachant que l'effectif doit rester pair.

```
def Selec(P,s):
    for I in P:
        if I==['m','m'] and random()<s:
            P.remove(I)
    if len(P)%2!=0:
        P.append(['m','m'])      # survivant
    return P
```

- Comment intégrer cette fonction dans le programme ?
- Faut-il modifier les autres fonctions ?

Solution

Les fonctions Pop, Dist et Pop_suivante sont inchangées.

- Que choisir pour s ?

Vers ressources

Intégration dans le programme :

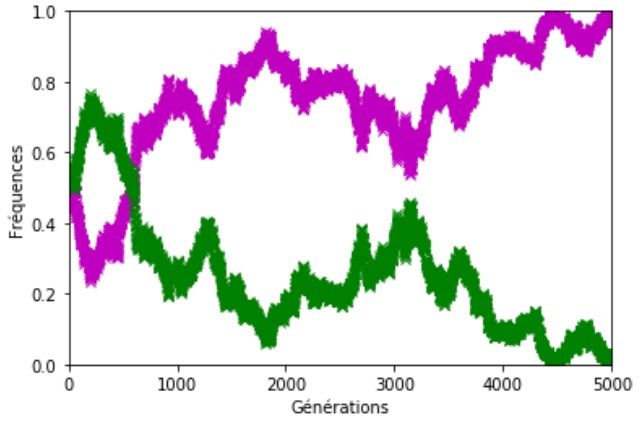
```
import matplotlib.pyplot as plt

n=int(input("Nombre de couples dans la population : "))
g=int(input("Nombre de générations : "))
s=float(input("facteur sélectif sur ['m','m'] : "))

plt.axis([-0.1,g,0,1])
plt.xlabel('Générations')
plt.ylabel('Fréquences')

P=Pop(n)
L=Dist(P)
M=[L[0]]
V=[L[1]]

for i in range(1,g+1):
    P=Pop_suivante(P)
    P=Selec(P,s)
    L=Dist(P)
    M.append(L[0])
    V.append(L[1])

abs=[k for k in range(g+1)]
plt.plot(abs,M,'mx')
plt.plot(abs,V,'gx')
plt.show()
```

Résultats sélection naturelle

Nombre de couples dans la population : 500

Nombre de générations : 5000

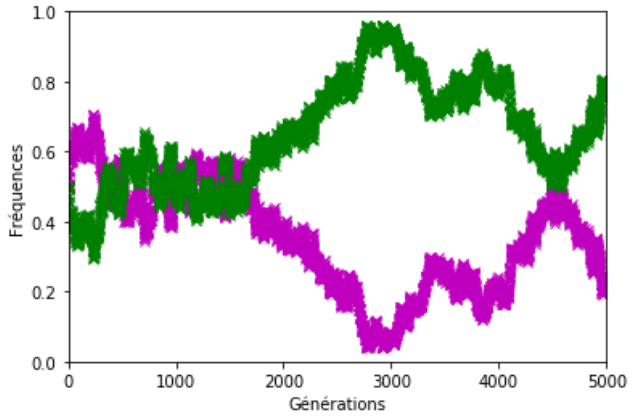
facteur sélectif sur ['m','m'] : 0.0001



Nombre de couples dans la population : 500

Nombre de générations : 5000

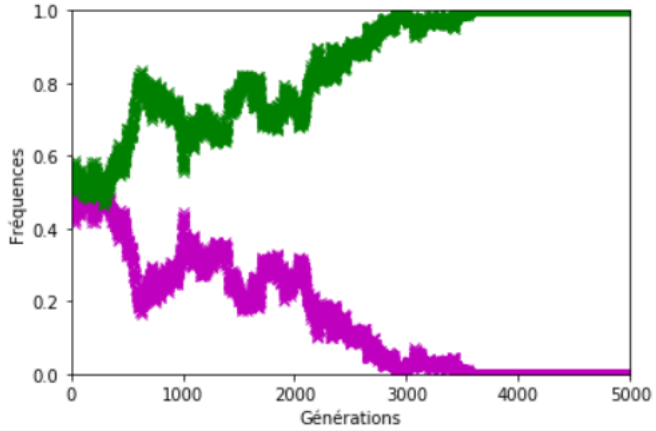
facteur sélectif sur ['m','m'] : 0.001



Nombre de couples dans la population : 500

Nombre de générations : 5000

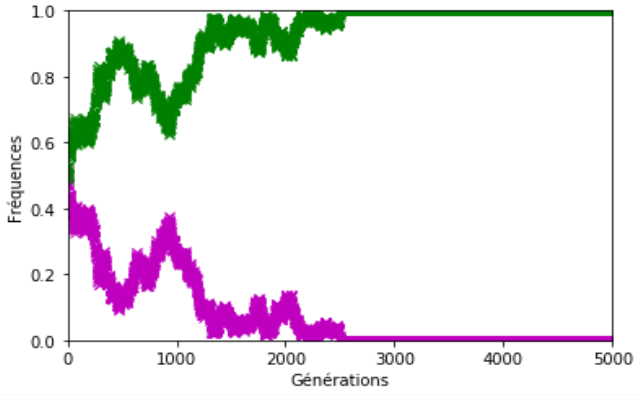
facteur sélectif sur ['m','m'] : 0.005



Nombre de couples dans la population : 500

Nombre de générations : 5000

facteur sélectif sur ['m','m'] : 0.01



Appariement non aléatoire

Les appariements ne sont plus équiprobables. On choisit de forcer l'appariement en priorité entre les individus ['m','m'] avec une fréquence c.

- Dans un premier temps, et avec une fréquence c, on choisit un individu ['m','m'] dans la population, s'il existe !
- Ensuite, on choisit un deuxième individu ['m','m'], s'il existe encore, et on reproduit le couple ainsi formé en créant les deux descendants.
- Dans les autres cas on procède à la reproduction des individus de façon équiprobable.
- On réitère le processus jusqu'à vider la population.

Quelle partie du programme ou des fonctions initiales faut–il modifier ?

Solution

Seule la fonction Pop_suivante est modifiée.

- Quelle(s) conséquence(s) sur l'évolution des fréquences alléliques ?
- Quelle valeur donner au coefficient d'appariement c ?

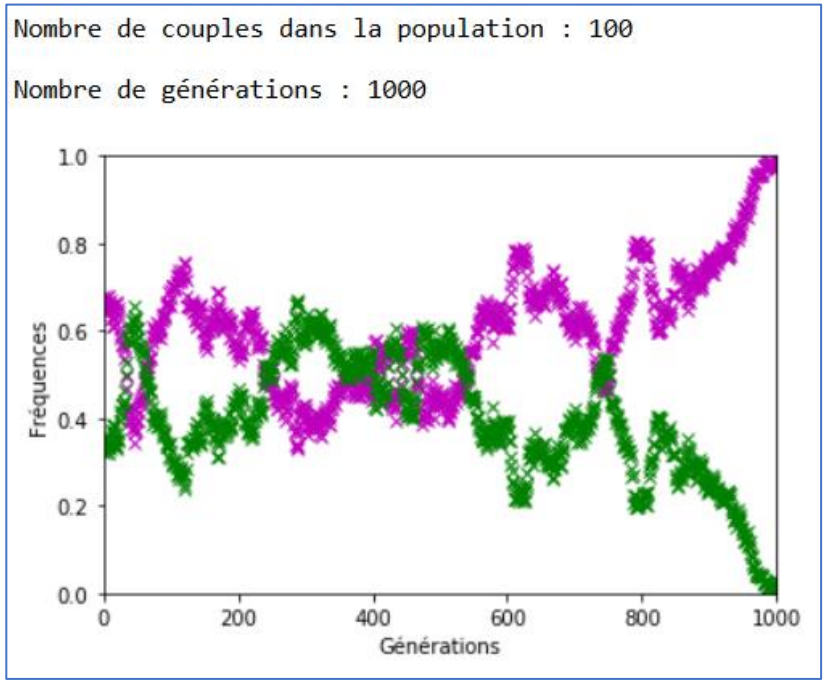
Vers ressources

```
def Pop_suivante(P,c):
    P_suivant=[]
    while P!=[]:
        if random()<c and ['m','m'] in P:
            P.remove(['m','m'])
            I_1=['m','m']
            if ['m','m'] in P:
                P.remove(['m','m'])
                I_2=['m','m']
                E_1=[choice(I_1),choice(I_2)]
                E_2=[choice(I_1),choice(I_2)]
                P_suivant.append(E_1)
                P_suivant.append(E_2)
            else:
                I_2=choice(P)
                P.remove(I_2)
                E_1=[choice(I_1),choice(I_2)]
                E_2=[choice(I_1),choice(I_2)]
                P_suivant.append(E_1)
                P_suivant.append(E_2)
        else:
            I_1=choice(P)
            P.remove(I_1)
            I_2=choice(P)
            P.remove(I_2)
            E_1=[choice(I_1),choice(I_2)]
            E_2=[choice(I_1),choice(I_2)]
            P_suivant.append(E_1)
            P_suivant.append(E_2)
    return P_suivant
```

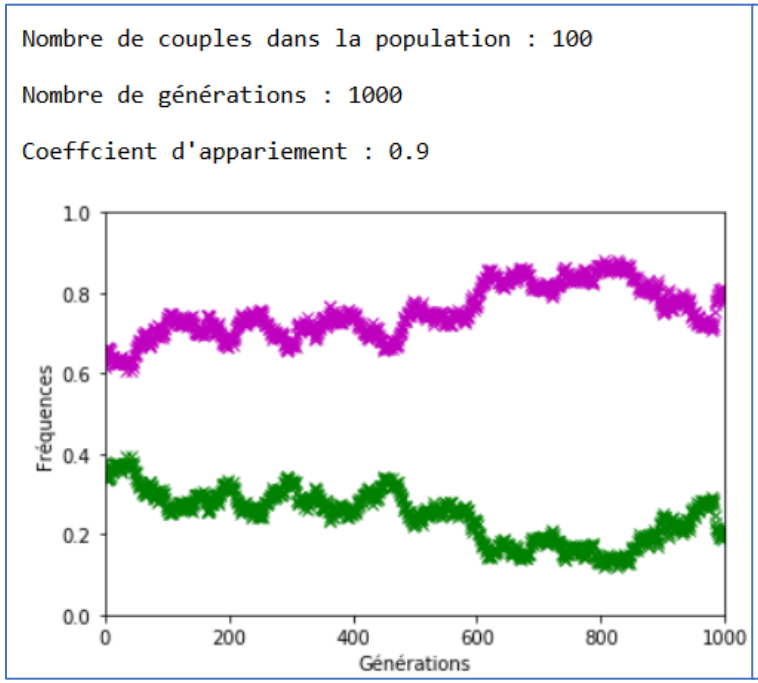
Comparaison avec et sans appariement aléatoire

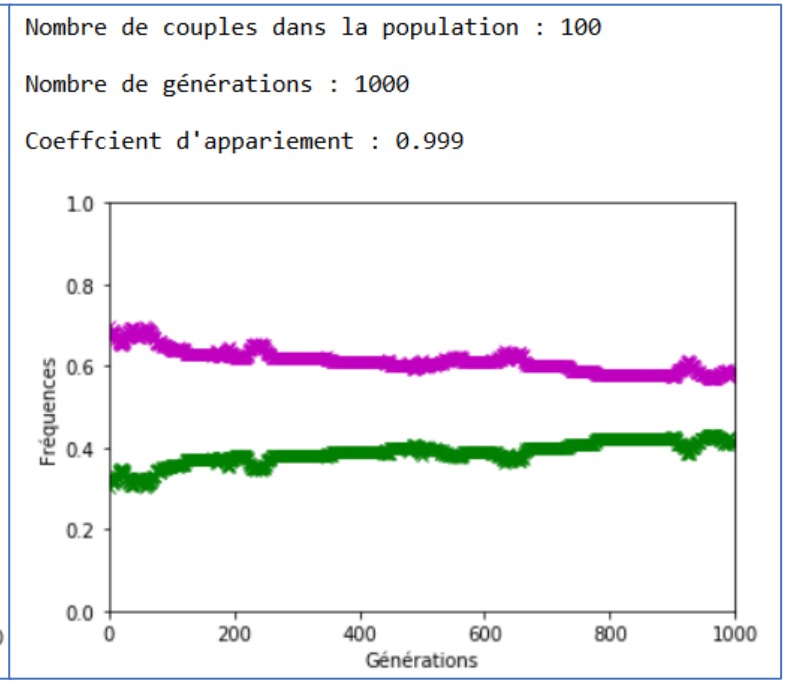
On choisit ici une population initiale avec **deux fois plus** d'allèles 'm' que d'allèles 'v'.

Appariement aléatoire



Appariement non aléatoire





Diminution de l'amplitude de la dérive génétique ? Et les phénotypes ?

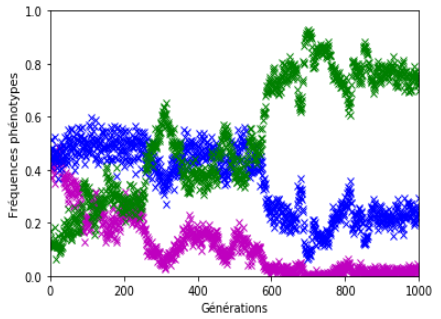
Du génotype au phénotype

La fonction « **Phéno** » traduit le génotype en phénotype ; **‘m’ et ‘v’ sont codominants** :

m//m en M
m//v en B
v//v en V

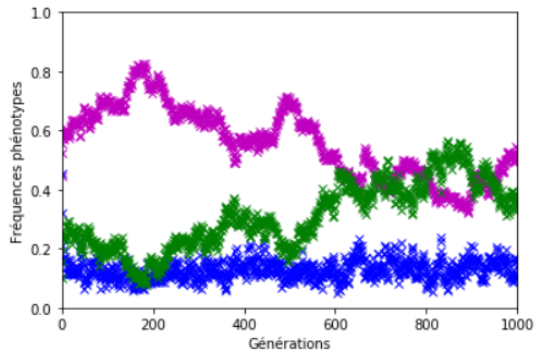
```
def Phéno(P):
    P_ph=[]
    for I in P:
        if I==['m','m']:
            P_ph.append('M')
        if I==['m','v'] or I==['v','m']:
            P_ph.append('B')
        if I==['v','v']:
            P_ph.append('V')
    return P_ph
```

```
Nombre de couples dans la population : 100

Nombre de générations : 1000

Coeffcient d'appariement : 0
Fréquences respectives des phénotypes 'M', 'B' et 'V' au départ : 0.415 0.465 0.12
```



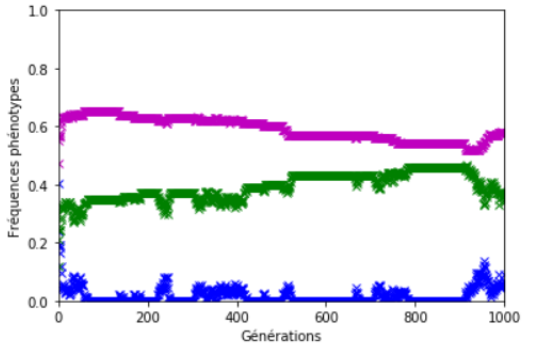

```
Fréquences respectives des allèles 'm' et 'v' : [0.1425, 0.8575]
```

```
Nombre de couples dans la population : 100

Nombre de générations : 1000

Coeffcient d'appariement : 0.9
Fréquences respectives des phénotypes 'M', 'B' et 'V' au départ : 0.445 0.455 0.1
```




```
Fréquences respectives des allèles 'm' et 'v' : [0.5825, 0.4175]
```

```
Nombre de couples dans la population : 100

Nombre de générations : 1000

Coeffcient d'appariement : 0.999
Fréquences respectives des phénotypes 'M', 'B' et 'V' au départ : 0.475 0.405 0.12
```




```
Fréquences respectives des allèles 'm' et 'v' : [0.615, 0.385]
```

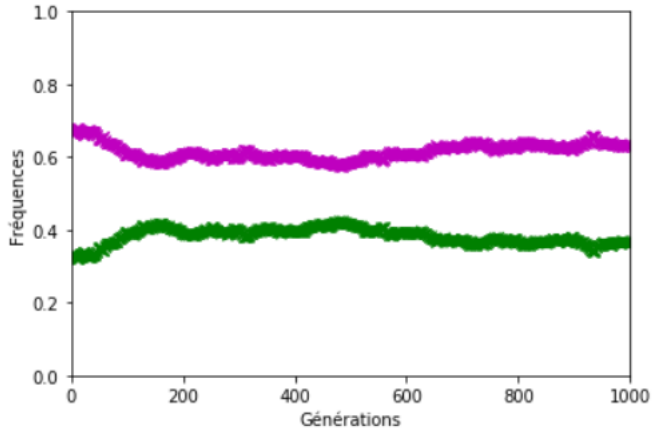
Le phénotype B (génotype m//v) tend à disparaitre.

Suite →

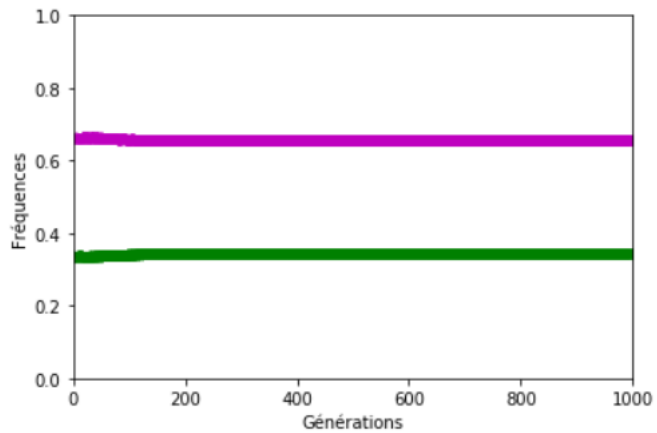
Les fréquences alléliques théoriques sont les suivantes : p=2/3 et q=1/3 (voir ci-contre).
Ce qui est ici vérifié avec ou sans appariement aléatoire mais un appariement fort estompe la dérive.

Dans le cadre du modèle de Hardy-Weinberg on obtient pour les phénotypes :
fmm=4/9, fvv=1/9, fmv=4/9.
Cet équilibre n'est pas vérifié en cas d'appariement fort (voir ci-dessous).

```
Nombre de couples dans la population : 5000

Nombre de générations : 1000

Coeffcient d'appariement : 0
```

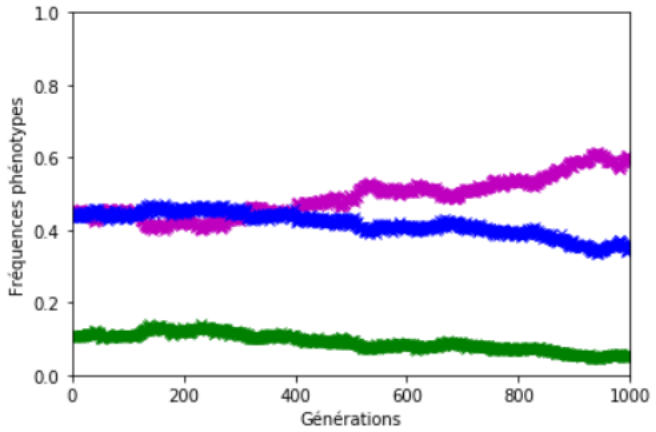



```
Nombre de couples dans la population : 5000

Nombre de générations : 1000

Coeffcient d'appariement : 1
```



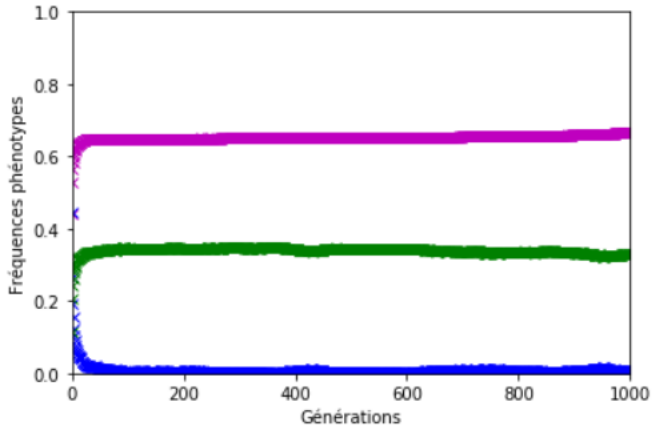
Représentation des phénotypes

```
Nombre de couples dans la population : 5000

Nombre de générations : 1000

Coeffcient d'appariement : 0
Fréquences respectives des phénotypes 'M', 'B' et 'V' au départ : 0.4512 0.4397 0.1091
```




```
Fréquences respectives des allèles 'm' et 'v' : [0.77235, 0.22765]
```

```
Nombre de couples dans la population : 5000

Nombre de générations : 1000

Coeffcient d'appariement : 1
Fréquences respectives des phénotypes 'M', 'B' et 'V' au départ : 0.4417 0.446 0.1123
```




```
Fréquences respectives des allèles 'm' et 'v' : [0.66765, 0.33235]
```

Un petit historique de la loi de Hardy- Weinberg

“Pedigree de brachydactylie de la famille Drinkwater"
extrait de *Mendélisme et relations avec les maladies* aux *Actes,* ***Punnet*** *(1908)*

« La brachydactylie est un ”cas simple de mendélisme” cette maladie est dominante ».

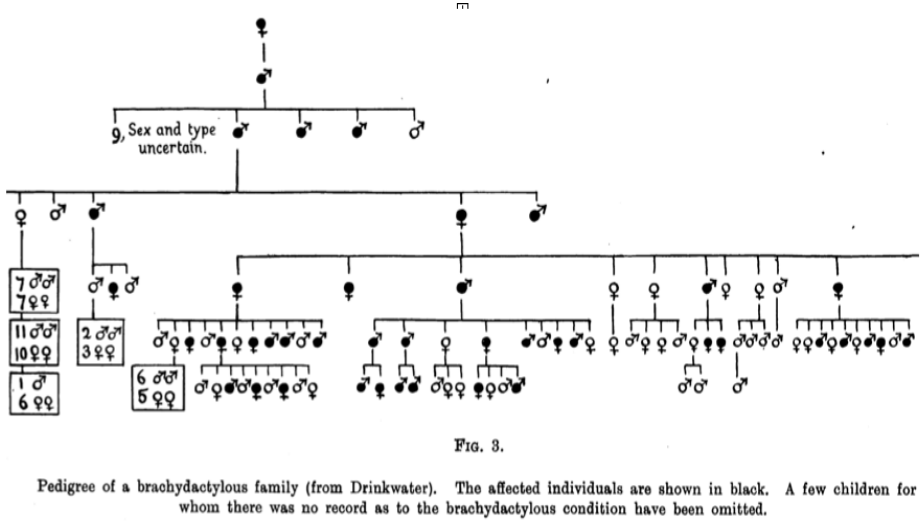


Fig. 3.

Pedigree of a brachydactylous family (from Drinkwater). The affected individuals are shown in black. A few children for whom there was no record as to the brachydactylous condition have been omitted.

George Udny Yule
(1871- 1951)
statisticien écossais

« Si la brachydactylie est dominante « au cours du temps, on pourrait s’attendre […] à obtenir trois personnes brachydactyles pour une normale. »

DISCUSSION ET CORRESPONDANCE
JULY 10, 1908 SCIENCE

Godfrey Harold Hardy

« Si la brachydactylie est dominante, la proportion de personnes brachydactyles de la deuxième génération est […] le double de la première génération; la proportion n'aura plus tendance par la suite à augmenter. […] »

Ressources

Propositions d’activités classe et programmes python sur le Drive suivant :

https://drive.google.com/drive/folders/1Uuu6uRnGVGMnUhwYHNHnqO8B7YUbAPtB?usp=sharing

FIN

Merci de votre attention

Pour plus d'information, vous pouvez nous contacter à l'adresse : j-louis.marcia@ac-creteil.fr
anne.juras@ac-creteil.fr
Renald-Philippe.Estavoyer@ac-creteil.fr