



# Transmission des maladies génétiques dans une population

Badr Saadioui  
Numéro de dossier : 26936

TIPE 2022  
Thème : Santé et prévention

# Plan

**01**

**Introduction**

**02**

**Problématique**

**03**

**Modèle de Wright-Fisher**

**04**

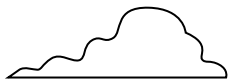
**Exemple de maladie  
héréditaire**

**05**

**Conclusion**

**06**

**Annexe**



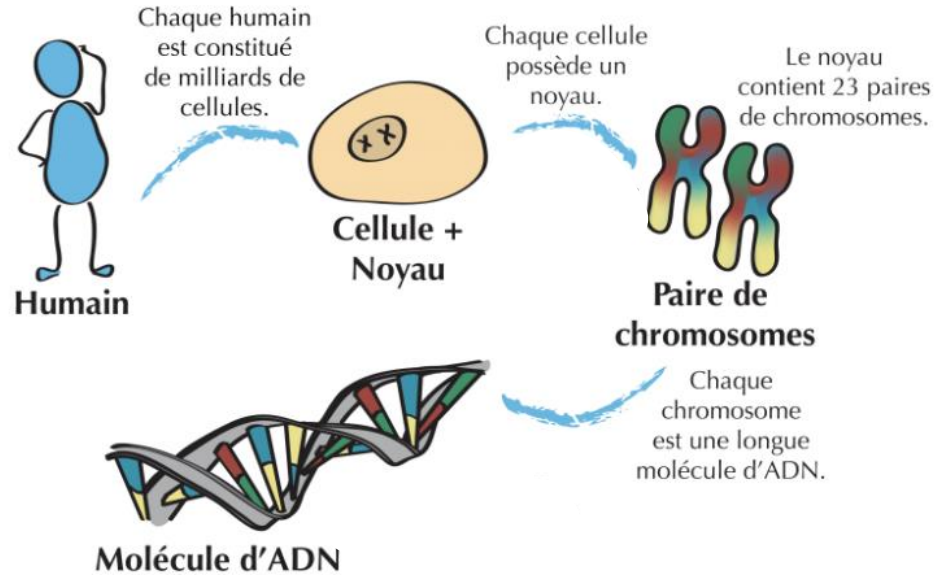
01

# Introduction



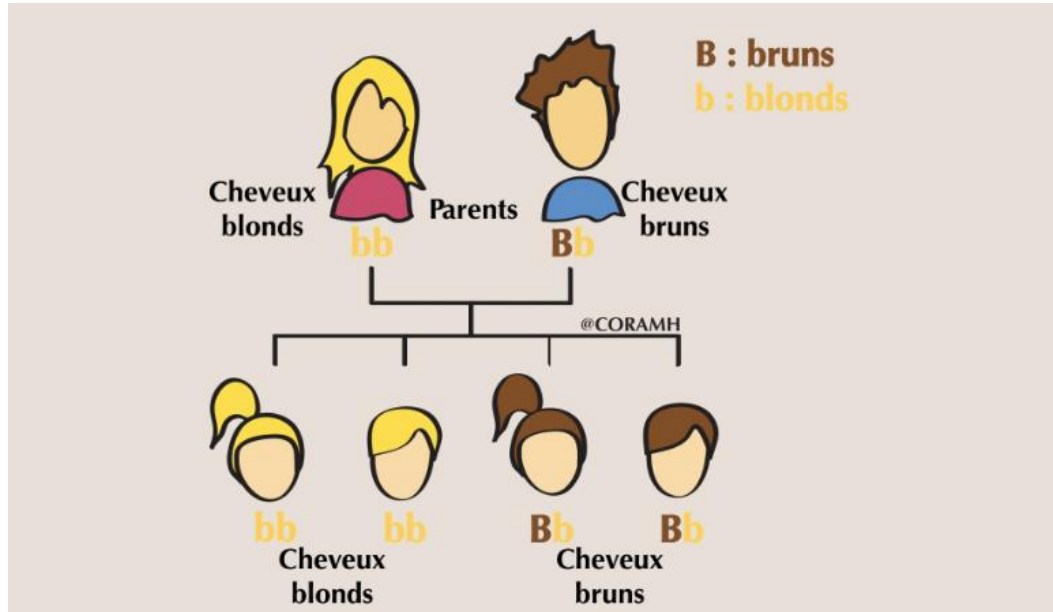
# Définitions

- La **génétique** est la science de l'hérédité.

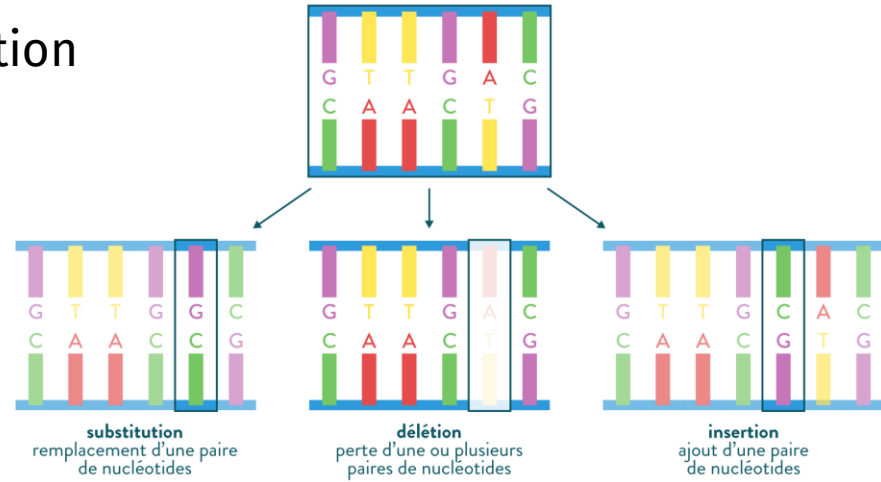


# Transmission des caractères héréditaires

- Chaque individu possède deux copies de chacun de ses gènes.
- L'**allèle** est le terme pour définir la copie du gène.



- Il arrive parfois qu'une mutation se produise dans l'ADN.



Lorsque ces mutations sont transmises d'une génération à l'autre, ces maladies génétiques sont dites héréditaires.

Ainsi on peut regrouper ces maladies héréditaires en deux grandes catégories selon le mode de transmission :

- **Les maladies héréditaires récessives**
- **Les maladies héréditaires dominantes**



02

# Problématique



# **Problématique**

**Peut-on décrire la transmission des allèles et l'évolution des fréquences alléliques au cours du temps ?**





03

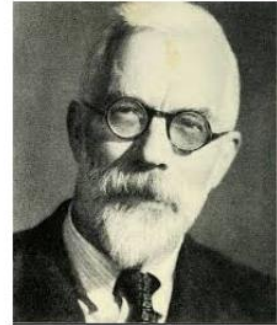
# Modèle de Wright-Fisher



le modèle **Wright-Fisher** décrit l'échantillonnage d'allèles dans une population **sans sélection**, **sans chevauchement** des temps de génération et **avec accouplement aléatoire**.



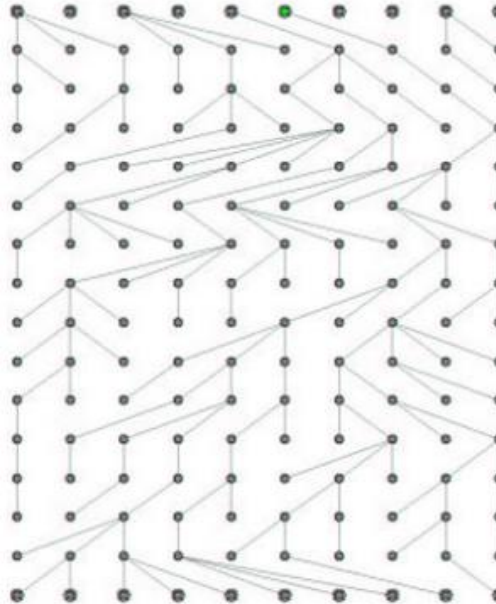
Sewall Wright



Ronald Fisher

## Modèle de Wright-Fisher

**N** individus se reproduisent aléatoirement. Chaque individu de la génération **n+1** choisit son parent uniformément au hasard dans les individus de la génération **n**.



Soit **A** une allèle dans une population donnée.

Le modèle de Wright-Fisher décrit l'évolution du nombre de cette allèle au fil du temps.

Soit **X(t)** une variable aléatoire qui décrit le nombre d'allèles **A** dans une population avec **N** individus à la génération **t** . L'ensemble des comptes possibles de l' allèle **A** c'est-à-dire les valeurs prise par **X(t)** sont **{0, 1 ..... 2N}**.

Alors:

$$X(t+1) \sim \text{Bin}(2N, \frac{X(t)}{2N})$$

C'est équivalent à :

$$\mathbb{P}(X(t+1) = k \mid X(t) = x) = \binom{n}{k} \left(\frac{x}{2N}\right)^k \left(1 - \frac{x}{2N}\right)^{2N-k}$$

## Théorème de l'espérance totale

Le théorème de l'espérance totale est une proposition de la théorie des probabilités affirmant que l'espérance de l'espérance conditionnelle de  $X$  sachant  $Y$  est la même que l'espérance de  $X$ .

on a alors le résultat suivant :

$$E(X) = E(E(X|Y))$$

## Moyenne

$$\begin{aligned} E(X_t) &= E(E(X_t \mid X_{t-1})) = E\left(2N \frac{X_{t-1}}{2N}\right) \\ &= E(X_{t-1}) = E(E(X_{t-1} \mid X_{t-2})) = E\left(2N \frac{X_{t-2}}{2N}\right) \\ &= E[X_{t-2}] = \dots = X_0 \end{aligned}$$

Nous voyons que la valeur attendue du processus de Wright-Fisher à tout moment n'est que le nombre d' allèles A dans la première génération.



04

# Exemple de maladie héréditaire : + la mucoviscidose

○



## La mucoviscidose, c'est quoi ?

La mucoviscidose est une maladie génétique qui touche en moyenne 1 enfant sur 3500. Elle entraîne une déficience pulmonaire d'où une mort prématurée de l'individu.



Comment la mucoviscidose peut-elle être si fréquente étant donné qu'elle entraîne une mort précoce et comment va évoluer la maladie dans les prochaines années ?



## Hypothèse

Les populations utilisés auront les caractéristiques suivantes :

- individus diploïdes (les individus possèdent deux exemplaires de chaque gène)
- individus hermaphrodites
- chaque individu descend de parents tirés aléatoirement dans la génération précédente
- les générations sont non-chevauchantes, c'est-à-dire qu'elles sont distinctes dans le temps et qu'il n'y a aucune interaction entre elles

## Mucoviscidose

La maladie en question est récessive.

Donc on observe trois cas possible :

allèles de l'individu	conséquence
MM	individu malade, stérile
SS	individu sain
SM	individu porteur sain

## Descendance

Voici quelque résultats :

Parent 1	Parent 2	Couple fertile ?	SS	SM	MM
SS	SS	OUI	1	∅	∅
SS	SM	OUI	0,5	0,5	∅
SS	MM	NON	∅	∅	∅
SM	SM	OUI	0,25	0,5	0,25
SM	MM	NON	∅	∅	∅
MM	MM	NON	∅	∅	∅

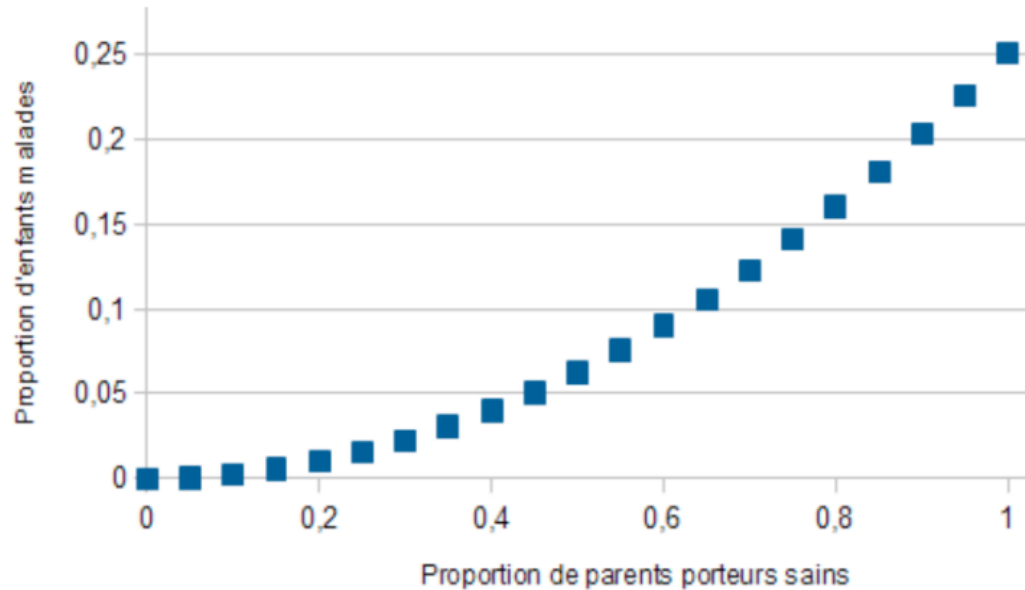
**Remarque :**

- Le nombre d'enfants malades d'une génération **n** ne dépend que du nombre de parents porteurs sains de la génération précédente.

On en déduit donc la formule suivante :

$$(fréquence\ MM\ à\ la\ génération\ n) = (fréquence\ de\ MS\ à\ la\ génération\ n - 1)^2 * \frac{25}{100}$$

Afin de mettre cette relation en évidence, on trace une courbe représentative de cette formule en fonction des proportions des porteurs sains.



## Proportion théorique des porteurs sains

- À partir de la loi précédente, on va calculer la proportion théorique de porteurs sains capables de transmettre la mucoviscidose.
- On note **SM** la fréquence des porteurs sains à la génération **n-1**, on a :

$$\frac{1}{3500} = (SM)^2 * \frac{25}{100}$$

$$\Leftrightarrow (SM)^2 = \frac{1}{3500} * \frac{100}{25}$$

$$\Leftrightarrow SM = \sqrt{\frac{1}{3500} * \frac{100}{25}} \simeq 3,4.10^{-2}$$

- Soit environ une personne sur trente est porteurs sains a la génération **n-1**

## **Propagation de la maladie dans le temps**

- Afin d'étudier la propagation de la maladie dans le temps on va utiliser des simulations sur des populations de tailles différentes et cela va permettre de collecter des données et d'en extraire des généralités.

## Petit échelle

Nous allons dans un premier temps réaliser des simulations de la propagation de la maladie dans des petites populations : 20 individus qui à la génération 0 ont chacun une chance sur deux d'être porteur sain. Voici un extrait du programme écrit en Python qui nous a permis de faire ces simulations :

```
12 #Initialisation
13 print("Initializing")
14 taillePop=20
15 freqM=0.5
16 G=0
17 SS,MM,MS=0,0,0
18 popP,popE=[[" "]*2 for i in range(taillePop)],[[" "]*2 for i in range(taillePop)]
19 for i in range(taillePop):
20     if randint(0,10)<=freqM*10:
21         popP[i][0]="S"
22         popP[i][1]="M"
23     else:
24         popP[i][0]="S"
25         popP[i][1]="S"
26
27 #Trouver un parent fertile
28 def parent(tab):
29     found=False
30     while not(found):#Bug potentiel si pop sterile (M;M = malade = stérile)
31         ind=randint(0,taillePop-1)
32         if tab[ind][0]=="S" or tab[ind][1]=="S":
33             found=True
34             return(ind)
```



```

50 #Comptage
51 def count(tab):
52     mm=0
53     ms=0
54     ss=0
55     for k in range(taillePop):
56         if tab[k][0]=="S" and tab[k][1]=="S":
57             ss+=1
58         elif tab[k][0]=="M" and tab[k][1]=="M":
59             mm+=1
60         else:
61             ms+=1
62     return(ss,mm,ms)
63
64
65 SS,MM,MS=count(popP)
66 print (G, " ",SS, " ",MM, " ",MS)

```

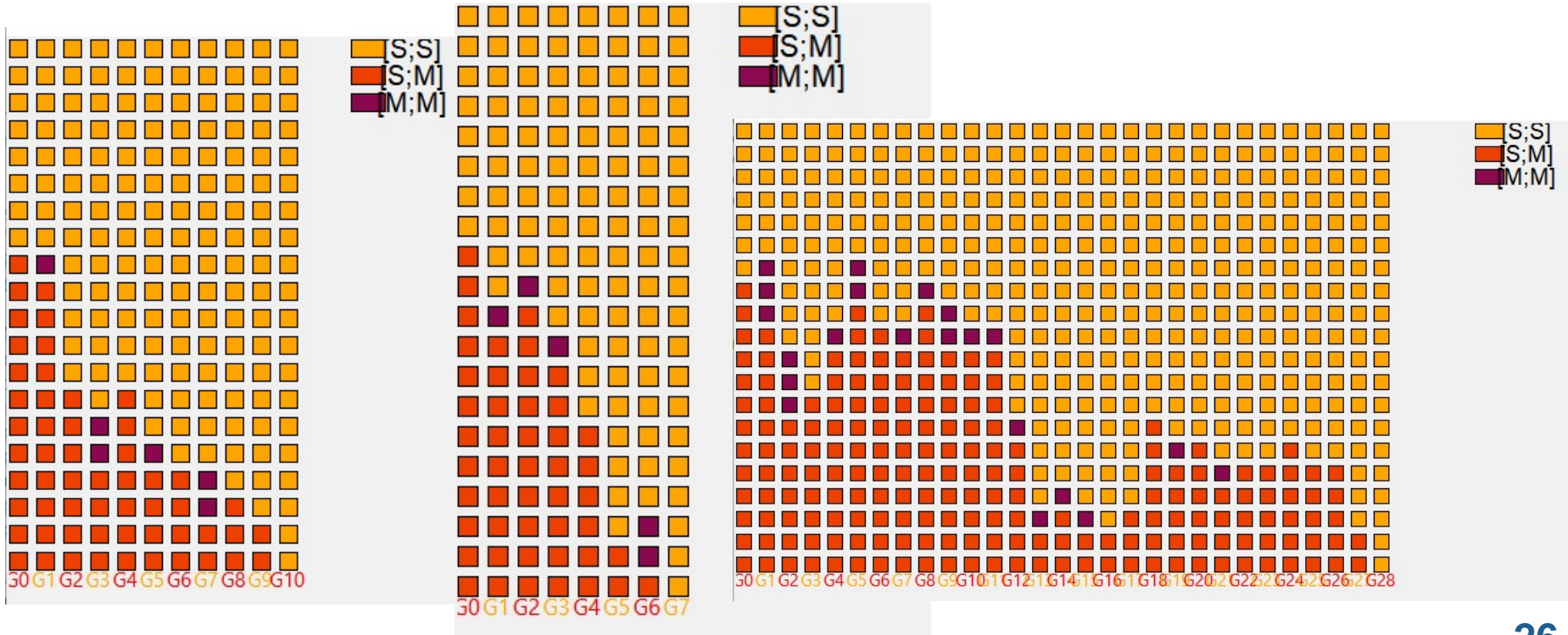
```

70 #Reproduction
71 while ( [ "S" , "M" ] in popP or [ "M" , "S" ] in popP ) and G<40:
72     G+=1
73     for j in range ( taillePop ): # determine la population des enfants
74         p1=parent (popP )
75         p2=parent (popP )
76         popE [ j ] [ 0 ] =popP [ p1 ] [ randint ( 0 , 1 ) ]
77         popE [ j ] [ 1 ] =popP [ p2 ] [ randint ( 0 , 1 ) ]
78
79     for k in range ( taillePop ): # les enfants deviennent les parents
80         for l in range (2):
81             popP [ k ] [ l ]=popE [ k ] [ l ]
82             popE [ k ] [ l ]= " "
83
84     SS ,MM,MS=count (popP )
85     print (G, " " ,SS , " " ,MM, " " ,MS

```

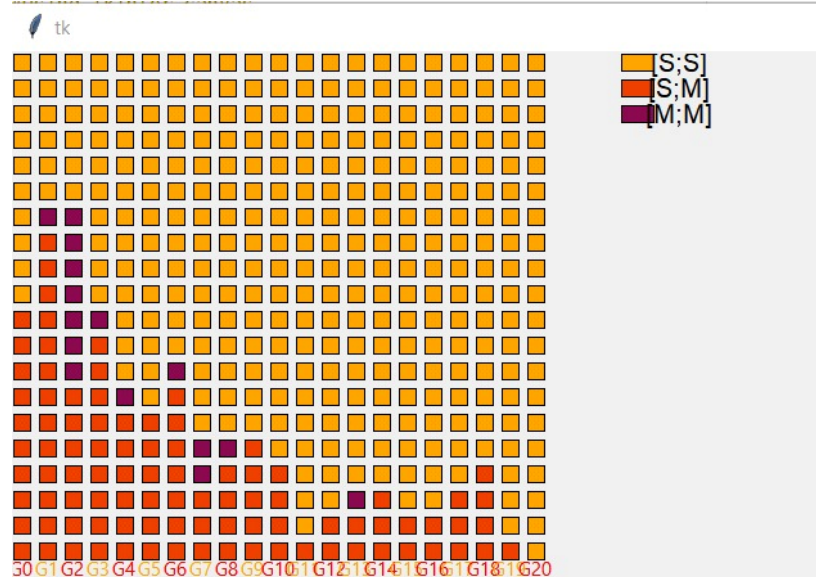
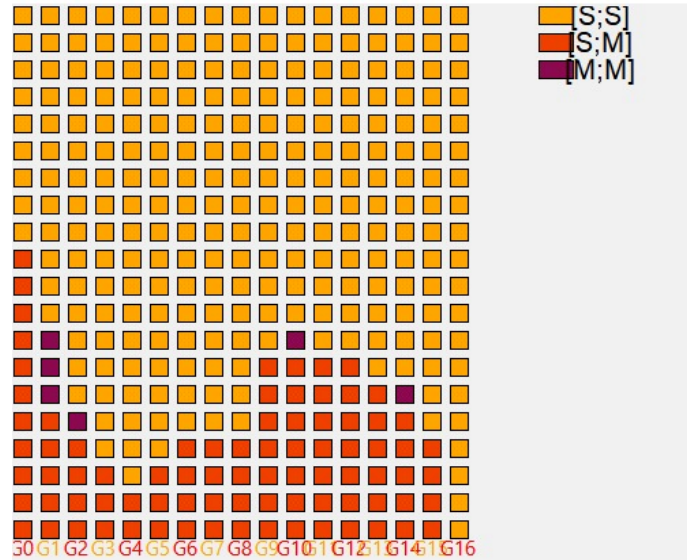
# Résultat

Voici les résultats que nous avons obtenus :



## Observation

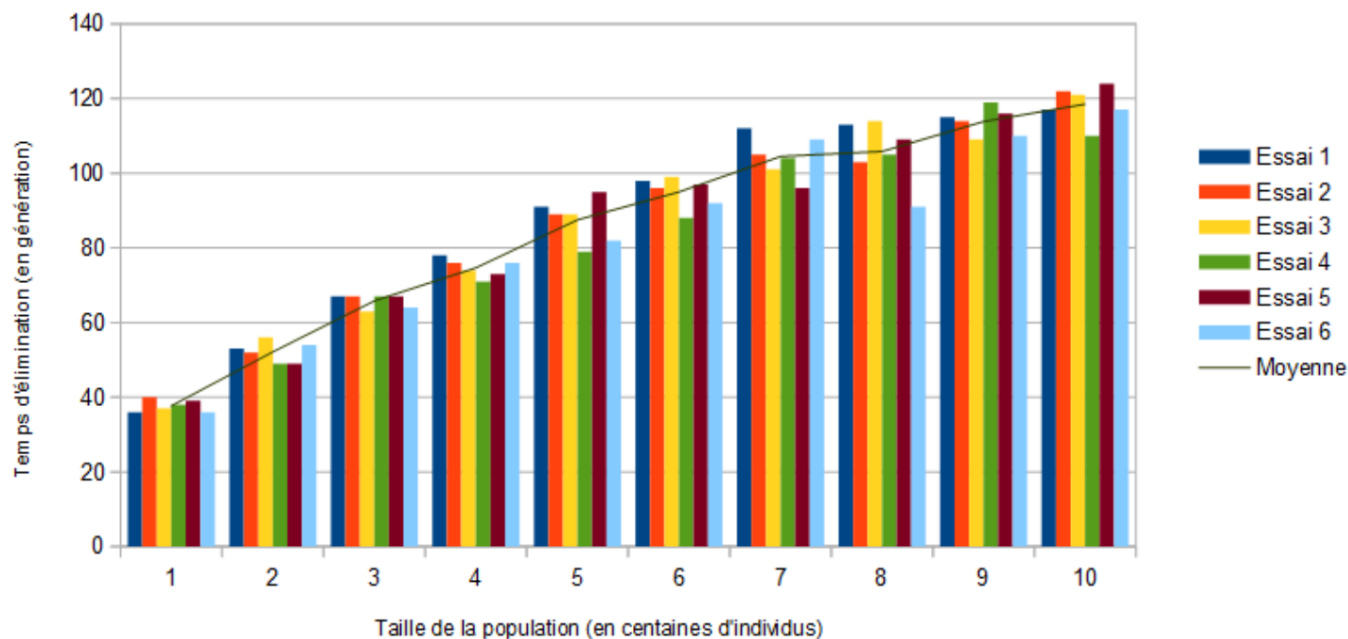
- Après plusieurs simulations : on observe que les résultats sont très différents mais qu'ils aboutissent tous à l'élimination de l'allèle M au bout d'un certain nombre de générations.



## Grande échelle et difficulté

- On souhaite faire de même à des populations de tailles comparables à la population mondiale actuelle, mais on ne dispose pas de moyens informatiques suffisamment puissants pour nous permettre de tester.
- On réalise des simulations sur des populations allant de 100 à 1000 individus. Suite aux simulations que nous avons effectuées, On va représenter graphiquement la courbe des moyennes des temps d'élimination de l'allèle M en fonction de la taille de la population.

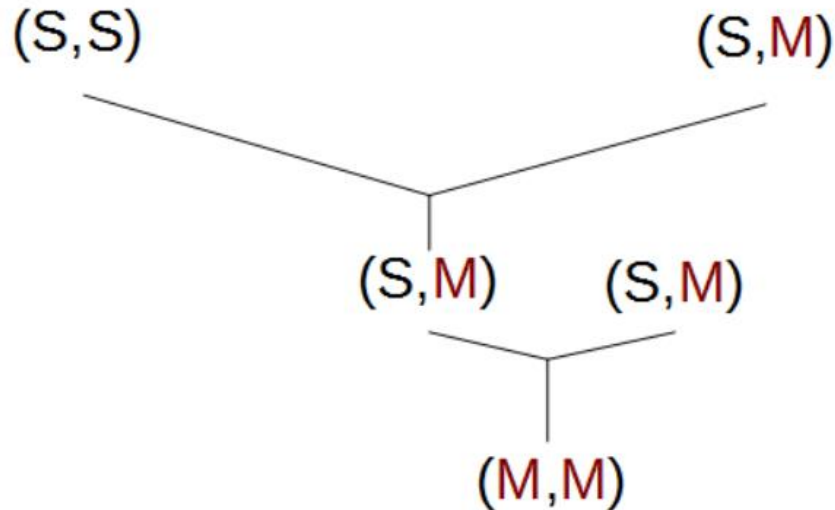
Voici la courbe obtenue :



**Diagramme du temps moyen d'élimination de l'allèle M dans une population en fonction de la taille de celle-ci**

## Scénario apocalyptique

On a observé que l'allèle M est éliminé au bout d'un certain nombre de générations, mais on souhaite savoir quel serait le temps minimal pour qu'une population soit entièrement malade.



On étudie une population de 100 individus dont 10% sont porteurs sains. Pour que la maladie se propage le plus rapidement dans la population, il faut que le nombre de porteurs sains soit maximal, or pour avoir des porteurs sains, il faut former le plus possible de couples constitués d'un individu porteur sain et d'un individu sain. On a donc représenté l'évolution de cette population par ce tableau :

génération	Individus (M,S)	Individus (S,S)	Individus (M,M)	Couples (S,S)/(S,S)	Couples (S,M)/(S,S)	Couples (S,M)/(S,M)
1	10	90	∅	40	10	∅
2	20	80	∅	30	20	∅
3	40	60	∅	10	40	∅
4	80	20	∅	∅	20	30
5	100	∅	∅	∅	∅	50
6	∅	∅	100	∅	∅	∅



Il faut donc au minimum 6 générations pour que la maladie envahisse totalement cette population. Voici la probabilité qu'un tel scénario se produise :

$$\begin{aligned} &0.5^{20} * 0.5^{40} * 0.5^{80} * [0.5^{40} + 0.5^{60}] * 0.25^{100} \\ &= 0.5^{140} * [0.5^{40} + 0.5^{60}] * 0.25^{100} \\ &\simeq 4 * 10^{-115} \end{aligned}$$

Nous n'avons pas pris en compte la probabilité qu'à chaque génération, tous les porteurs sains s'accouplent avec des individus sains si les couples sont formés aléatoirement. Ainsi, ce résultat est en réalité plus faible encore.



05

# Conclusion



**Merci pour  
votre attention**



06



**Annexe**



```

1
2 from random import *   #Beaucoup de modules !...
3 from math import *
4 from tkinter import *
5 from time import *
6
7 def main():
8     Population = Tk()   #Using Tkinter canvas
9     w = Canvas(Population, width=1000, height=520)
10    w.pack()
11
12    #Initialisation
13    print("Initializing")
14    taillePop=20
15    freqM=0.5
16    G=0
17    SS,MM,MS=0,0,0
18    popP,popE=[[" "]*2 for i in range(taillePop)],[[" "]*2 for i in range(taillePop)]
19    for i in range(taillePop):
20        if randint(0,10)<=freqM*10:
21            popP[i][0]="S"
22            popP[i][1]="M"
23        else:
24            popP[i][0]="S"
25            popP[i][1]="S"
26

```

```

26
27 #Trouver un parent fertile
28 def parent(tab):
29     found=False
30     while not(found):#Bug potentiel si pop sterile (M;M = malade = stérile)
31         ind=randint(0,taillePop-1)
32         if tab[ind][0]=="S" or tab[ind][1]=="S":
33             found=True
34             return(ind)
35
36 #Tkinter Update
37 def update(g,a,b,c):
38     for i in range(a): #<canvasName>.create_rectangle(<Xa>,<Ya>,<Xb>,<Yb>,fill=<"color">)
39         w.create_rectangle(20*g+2, 20*i+2, 20*g+15, 20*i+15, fill="orange", activewidth=5,
activeoutline="black")#orange clair
40     for i in range(a,b):
41         w.create_rectangle(20*g+2, 20*i+2, 20*g+15, 20*i+15, fill="DeepPink4", activewidth=5,
activeoutline="black")#noir
42     for i in range(b,c):
43         w.create_rectangle(20*g+2, 20*i+2, 20*g+15, 20*i+15, fill="orangered2", activewidth=5,
activeoutline="black")#orange sombre
44     c="orange"
45     if g%2==0:
46         c="red"
47     w.create_text(20*g+8, 20*(i+1)+2,fill=c, text="G{}".format(g))
48     #sleep(1)

```

```

50 #Comptage
51 def count(tab):
52     mm=0
53     ms=0
54     ss=0
55     for k in range(taillePop):
56         if tab[k][0]=="S" and tab[k][1]=="S":
57             ss+=1
58         elif tab[k][0]=="M" and tab[k][1]=="M":
59             mm+=1
60         else:
61             ms+=1
62     return(ss, mm, ms)
63
64
65 SS,MM,MS=count(popP)
66 update(0, SS, SS+MM, SS+MM+MS)
67

```

```

68 #Reproduction
69 while (["S","M"]in popP or ["M","S"] in popP) and G<40:#Autofecondation possible
70     G+=1
71     for j in range(taillePop):
72         p1=parent(popP)
73         p2=parent(popP)
74         popE[j][0]=popP[p1][randint(0,1)]
75         popE[j][1]=popP[p2][randint(0,1)]
76
77     for k in range(taillePop):#Les enfants deviennent les parents de la generation G+1
78         for l in range(2):
79             popP[k][l]=popE[k][l]
80             popE[k][l]=" "
81
82     SS,MM,MS=count(popP)
83     update(G,SS,SS+MM,SS+MM+MS)
84     #print(G," ",SS," ",MM," ",MS)
85     #print(popP)
86     #update(0,5,8,20)#Pour tester...resultat: update() marche, donc c'est count() qui foire
87 for i in range(3):
88     color="DeepPink4"
89     tx="[M;M]"
90     if i==0:
91         color="orange"
92         tx="[S;S]"
93     elif i==1:
94         color="orangered2"
95         tx="[S;M]"
96     w.create_rectangle(22*G+35, 20*i+2, 22*G+60, 20*i+15, fill=color, activewidth=5,
activeoutline="black")
97     w.create_text(22*G+80, 20*i+8,fill="black", text=tx,font="Helvetica")
98     w.mainloop()
99
100 main()

```