

**EM : Expliquer l'évolution de la pigmentation chez la phalène du bouleau
(Livre p87, documents 2 à 4)**

TP10 Comment expliquer l'origine de la biodiversité?

Partie 1: Comment évoluent les proportions d'allèles d'un gène au fil des générations?

A- Nous partirons ici du principe qu'aucun allèle n'offre d'avantage par rapport à un autre. 50mn

Activités et déroulement des activités	barème
Ressources	
<i>Sacs roses et bleus en tissu(4 pour chaque binome: deux bleus, deux roses) Perles de couleurs en nombre conséquent.(4 couleurs) Un ordinateur et un tableur Un environnement et un éditeur Python sur les machines Une imprimante</i>	
Étape 1 : Concevoir une stratégie pour résoudre une situation problème	
Proposer, en utilisant le matériel mis à disposition, la démarche qui vous permettrait de répondre à la problématique. La démarche est imposée, exposez la clairement en expliquant le rôle de chaque aspect du modèle. Ne pas oublier d'expliquer : - la façon de reproduire le paramètre aléatoire du passage des allèles d'une génération à l'autre lors de la reproduction. - comment évaluer, pour chaque génération, la proportion de chaque allèle présent.	2
Étape 2 : Mettre en œuvre un modèle pour obtenir des résultats exploitables	
Réaliser le modèle proposé Observer aussi l'effet de la taille de la population sur la dérive génétique grâce à la simulation disponible dans le logiciel évolution.exe présent dans le dossier	
Étape 3 : Présenter les résultats pour les communiquer	
Construire une représentation adaptée qui permet de mettre en évidence l'évolution des proportions des allèles de votre modèle. (construire le graphique représentant l'évolution de la proportion de chaque allèle au cours des générations.)	4
Étape 4 : Exploiter les résultats obtenus pour répondre au problème	
Analyser vos représentations et observations pour répondre à la problématique grâce à un texte concis.	1

Un modèle analogique proposé: exposé par le professeur

Soit une population fermée, isolée du reste du monde, d'individus possédant chacun 2 des 3 allèles a1, a2 et a3 d'un gène:(c'est une espèce diploïde!!)

A=(a1//a2) #les génotypes des individus du village

B=(a2//a3) C=(a3//a1) D=(a1//a1) E=(a2//a2) F=(a3//a3)

ou en python :

A=['a1','a2'] #les génotypes des individus du village

B=['a2','a3']

C=['a3','a1']

D=['a1','a1']

E=['a2','a2']

F=['a3','a3']

à vos cotillons !

Les allèles seront figurés par des perles de couleurs différentes, placés par deux(à expliquer) dans des sacs de tissus roses(femmes) et bleus(hommes).

Toujours de manière **aléatoire**: choisir, à l'aveugle, **4 individus reproducteurs**, déterminer leur **sexes respectifs**(2 femmes et 2 hommes), fonder **2 couples**, ils auront respectivement **3 enfants chacun**, les allèles reçus par chaque enfant doivent être tirés au sort.

Répéter cette opération **durant x générations** en comptabilisant le nombre de chaque allèle obtenu dans la génération dans un tableur afin de réaliser un graphique de l'évolution du pourcentage de chaque allèle dans la génération.

Afin de montrer la diversité potentielle des résultats, nous travaillerons en projet collaboratif :

Cliquez sur le lien envoyé par mail sur l'ENT, utilisez ensuite : [identifiant](#) prenom.nom@monlycee.net et mot de passe d'accès à l'ENT/

Modèle numérique : à votre python !

ouvrir le fichier 'progDeriveModeleViable' avec un éditeur et digérez le pour le comprendre... voir 'ANNEXE le programme python'.

Lancer le programme et répondez aux demandes.

Une fois le programme python utilisé, le fichier « derive.csv » est créé, l'ouvrir avec openalc et configurer le traitement ainsi :

	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	St
1	frequence_allele_a1	33	33	33	33	33	16	25	33
2	frequence_allele_a2	58	58	58	50	58	66	66	66
3	frequence_allele_a3	8	8	8	16	8	16	8	0
4	Générations	0	1	2	3	4	5	6	7

L'enregistrer ensuite avec un nom choisi à l'endroit de votre choix en format .odf.

A-1 Traiter ensuite le tableau afin d'en faire un graphique utilisable pour expliquer : « Comment évoluent les proportions d'allèles d'un gène au fil des générations lorsqu'ils ont la même valeur sélective »?(consigne A-2!!)

Pistes :

- la confrontation des simulations de chaque groupe peut avoir un grand intérêt à condition de respecter une certaine rigueur scientifique.

- Faire varier grandement les effectifs de votre modèle peut apporter une information intéressante.

→ Sur le graphique python : la courbe concernant a1 est de couleur bleue, a2 de couleur verte et a3 en rouge.

B- Comment l'environnement fait pression sur l'évolution de la biodiversité ?

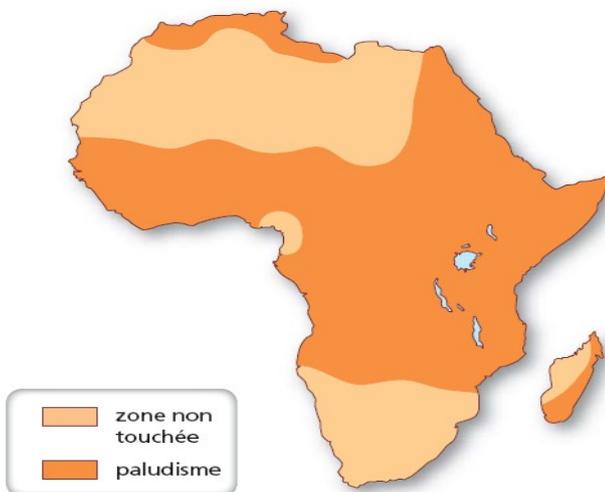


Les girafes naissent avec un cou de longueur légèrement différente les unes des autres. Cette variation héréditaire est aléatoire.

En revanche, il n'y a plus rien d'aléatoire dans le fait que la girafe ayant le cou le plus long accède à plus de feuilles. Mieux nourrie, elle aura plus de chances de se reproduire et de donner une descendance à laquelle elle va transmettre cette particularité du cou le plus long.

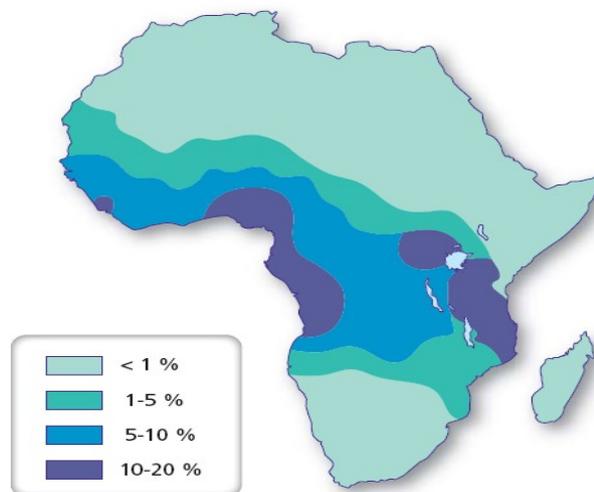
De cette manière, la population évolue en se modifiant en fonction de son environnement. C'est Darwin, le premier, qui décrit ce phénomène et le nomma « sélection naturelle ».

Le paludisme est une maladie infectieuse due à un parasite unicellulaire (*Plasmodium*) qui se développe en partie dans les hématies. Ce parasite est transmis à l'Homme lors d'une piqûre par un moustique. Le paludisme est l'un des principaux fléaux mondiaux, tuant chaque année entre 1,5 et 2,5 millions d'enfants de moins de cinq ans.



Doc. 3 Distribution du paludisme en Afrique.

La drépanocytose est une maladie génétique (voir unité 4, p. 52). Cette maladie se développe uniquement chez les individus portant les deux allèles S responsables de la forme S de l'hémoglobine. Des études cliniques ont montré que la présence d'hémoglobine S dans les hématies réduisait la sensibilité au *Plasmodium*. Elle atténue ainsi la gravité des crises de paludisme.



Doc. 4 Distribution de la fréquence de l'allèle S de la drépanocytose en Afrique.

Pour répondre à cette problématique, vous pourrez, grâce aux documents de la page 88 ou de la page 89:
– Argumenter l'influence de l'environnement sur la sélection de certains allèles.

Lancer de nouveau le programme Python « progSelectNatModeleViable »: attribuer à chaque d'allèle une valeur sélective.

Le modèle part d'un principe de codominance... si a1 a un indice de 0,8 et a2 de 0,6 → a1//a1 a une valeur sélective de 0,8 et a1//a2 de 0,7 et a2//a2 de 0,6.....

Puis on applique l'indice de valeur sélective : Lire le code pour essayer de comprendre comment j'ai décidé de modéliser la sélection naturelle.

B- 1- Expliquer le principe de modélisation utilisé pour simuler une sélection naturelle dans ce modèle numérique.

B-2- Expliquer le déroulement de la sélection naturelle

EM : Partie 3: Comment apparaissent de nouvelles espèces?

Ressources :

Les facteurs influençant la biodiversité comme les mutations, la dérive génétique ou la sélection naturelle vont avoir une influence sur l'évolution d'une espèce mais pas forcément pour faire naître deux espèces à partir d'une seule...

Une espèce supplémentaire est définie si un **nouvel ensemble s'individualise**. Pour cela :

- il faut qu'**au** moins deux groupes soient isolés, les individus de ces groupes ne pourront plus se reproduire. **Deux solutions :**

1- la spéciation **allopatrique** : l'isolement est de nature géographique, les individus ne sont plus en contact.

2- La spéciation **sympatrique** : l'isolement est de nature reproducteur, les individus restent en contact sans pouvoir se croiser en donnant une descendance fertile. (ex : isolement temporel : les gamètes ne sont pas émis en même temps.)

Cet isolement permettra :

1- si l'on se place dans une des deux populations isolées, de subir des mutations, donc de voir apparaître de nouveaux allèles ou gènes et de les garder dans SA population sans les donner à l'AUTRE. Petit à petit, les deux populations accumuleront les différences du fait du hasard des mutations. Les **modifications génétiques** (mutations aléatoires, duplications, ruptures de chromosomes ou transferts...) vont se produire de façons différentes dans chaque population isolée.

2- si les deux(ou plus) environnements qui accueillent les populations sont différents peuvent engendrer des **pressions différentes(voir inverses) sur le même allèle**. Un individu présentant le caractère innovant sera favorisé dans un environnement 1 et pourra être défavorisé dans un environnement 2 : les **sélections naturelles sont différentes**.

3- La **dérive génétique étant aléatoire, elle se déroulera forcément de façons différentes** dans chacune des populations isolées, même si les conditions y sont semblables.

→ On parlera de spéciation lorsque les individus des deux populations ne pourront plus s'hybrider en donnant une descendance fertile...(mais on a vu que ce n'était pas si simple...)

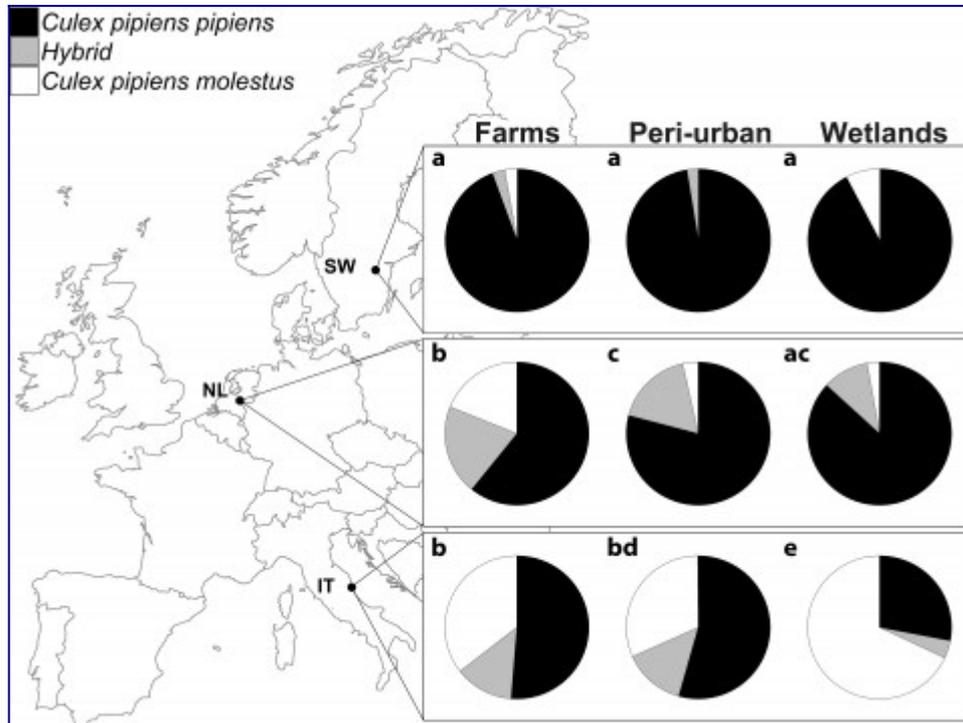
Pour répondre à cette problématique, vous pourrez, grâce aux documents p 92-93 du livre LLS:

- Expliquer les mécanismes possibles d'isolement reproducteur dans cette population d'escargots.
- En vous aidant des documents, rédiger un avenir possible des populations isolées...

*Pour ceux que ça intéresse :

On a longtemps cru que les moustiques du métro de Londres et ceux de la surface avaient la même origine et qu'une spéciation avait eu lieu en moins d'un siècle grâce à la séparation en deux de la population de surface d'origine lors de la construction du métro. Une partie de la population a élu domicile dans le métro, le reste est resté en surface. Les deux populations ainsi isolées auraient donc évolué chacune dans leur coin pour donner naissance à deux espèces distinctes ...

Deux études plus récentes ont montré que les hybrides entre *pipiens* et *molestus* existent bel et bien dans toute l'Europe, même s'ils sont plus fréquents aux basses latitudes qu'aux hautes latitudes (Rudolf et al., 2013; Vogels et al., 2016).



Répartition des populations de *Culex pipiens* dans trois pays

Auteur : Vogels et al.

1- il existe finalement des individus issus du croisement entre pipiens et molestus → nous n'avons donc pas affaire à deux populations isolées génétiquement car elles peuvent se reproduire ensemble → il s'agit de deux sous-espèces et non de deux espèces différentes

2- molestus existait déjà bien avant le métro de Londres, il existe des populations un peu partout en Europe → le molestus du métro de Londres présente plus de similitudes génétiques avec ses cousins européens qu'avec le pipiens de Londres → on a interprété qu'il venait donc d'un ancien cousin molestus qui était rentré dans le métro après sa fabrication.

ANNEXE Le programme python

```
def main():
    pass
if __name__ == '__main__':
    main()

from random import* # on aura besoin d'aléatoire...

# on affecte toutes les variables utiles par la suite...

A=['a1','a2'] #les génotypes des individus du village
B=['a2','a3']
C=['a3','a1']
D=['a1','a1']
E=['a2','a2']
F=['a3','a3']
Village=[A,B,C,D,E,F] # le village
Villagen=[] #le nouveau village en fin de sélection des reproducteurs
RESULTATS=[] #résultats à remplir sous forme d'une liste de Nbgeneration listes
#à 6 listes de deux éléments, les deux allèles

Nbgenerations=int(input('Sur combien de générations voulez vous travailler?'))
# on travaillera d'abord sur x générations

lista1=[] #pour générer les listes vides
lista2=[] #de chaque allèle à remplir ensuite au cours des générations
lista3=[]
GEN=[] # idem pour les abscisses générations du graph

#on définit les fonctions permettant, les choix des 4 reprod parmi les 6 ind,
# la génèse de la nouvelle génération à partir des 4 reproducteurs,
# le traitement des données obtenus au bout de x générations pour obtenir des listes
# la mise en fichier csv afin de pouvoir l'exploiter en tableur.

def choixreprod(Village): # tourne en fonction du village initial
    rep1=randint(0,5)
    rep2=randint(0,5)
    while rep1==rep2:
        rep2 = randint(0,5)
    rep3 = randint(0,5)
    while rep3==rep2 or rep3==rep1:
        rep3 = randint(0,5)
    rep4 = randint(0,5)
    while rep4==rep3 or rep4==rep2 or rep4==rep1:
        rep4 = randint(0,5)
    REP1=Village[rep1]
    REP2=Village[rep2]
    REP3=Village[rep3]
    REP4=Village[rep4]
    Villagen=[REP1, REP2,REP3,REP4] # crée un nouveau village de reproducteur
#en fonction du village intial
    print('les couples reproducteurs tirés au sort seront {}, {} et {}, {}'.format(REP1, REP2, REP3, REP4))
    return Villagen # renvoi le nouveau village de reproducteurs

def cyclereprod(Villagen): # utilise
```

```

#print(Villagen) #test de conservation de la variable Villagen
a=randint(0,1) # on peut le mettre directement à la place de a ligne d'en dessous
#mais c'est plus pédagogique ainsi....
allele1=Villagen[0][a] # enfants du couple 1 du village
a=randint(0,1)
allele2=Villagen[1][a]
Enf1=[allele1,allele2]
a=randint(0,1)
allele1=Villagen[0][a] # enfants du couple 1 du village
a=randint(0,1)
allele2=Villagen[1][a]
Enf2=[allele1,allele2]
a=randint(0,1)
allele1=Villagen[0][a] # enfants du couple 1 du village
a=randint(0,1)
allele2=Villagen[1][a]
Enf3=[allele1,allele2]

allele1=Villagen[2][randint(0,1)] # enfants du couple 2 du village
allele2=Villagen[3][randint(0,1)]
Enf4=[allele1,allele2]
allele1=Villagen[2][randint(0,1)]
allele2=Villagen[3][randint(0,1)]
Enf5=[allele1,allele2]
allele1=Villagen[2][randint(0,1)]
allele2=Villagen[3][randint(0,1)]
Enf6=[allele1,allele2]

Village=[Enf1,Enf2,Enf3,Enf4,Enf5,Enf6]
RESULTATS.append(Village)
print('A la {}eme génération, les enfants obtenus sont {}'.format(G,Village))

return Village

```

```

def traitementresultats(RESULTATS):
    for G in range (0, Nbgenerations):
        a1=0
        a2=0
        a3=0
        for g in range(0,6):
            for a in range(0,2):
                #print('{} {} {} test {}'.format(G,g,a,RESULTATS[G][g][a])) #test
                if RESULTATS[G][g][a]=='a1':
                    a1=a1+1
                elif RESULTATS[G][g][a]=='a2':
                    a2=a2+1
                elif RESULTATS[G][g][a]=='a3':
                    a3=a3+1

            a1=int(a1/12*100) # on convertit en pourcentage
            a2=int(a2/12*100)
            a3=int(a3/12*100)
            #print (a1, a2, a3, G) # test pour voir l'évolution en temps réel

        lista1.append(a1) #pour générer les listes de quantité
        lista2.append(a2) #de chaque allèle au cours des générations associé à la génération

```

```
lista3.append(a3)
GEN.append(G)
```

```
return lista1
return lista2
return lista3
return GEN
```

```
def miseenfichiercsv(lista1, lista2, lista3, GEN):
    f=open('D:/essaipython/dérive.csv', 'w') # lien vers un fichier on l'affecte la variable f
    f.close()
    f=open('D:/essaipython/dérive.csv', 'a') # lien vers un fichier on l'affecte la variable f
    lista1=str(lista1)
    lista2=str(lista2)
    lista3=str(lista3)
    GEN=str(GEN)
    f.write('frequence_allele_a1')
    f.write(lista1)
    f.write('\n') # permet de séparer les lignes dans le csv, à la ligne
    f.write('frequence_allele_a2')
    f.write(lista2)
    f.write('\n')
    f.write('frequence_allele_a3')
    f.write(lista3)
    f.write('\n')
    f.write('Génération')
    f.write(GEN)
    f.close()
```

```
# on lance tout
```

```
for G in range(Nbgenerations):
    V=choixreprod(Village) # on attribue à V la liste des nouveaux reproducteurs
    Village=cyclereprod(V) #on fait des enfants avec les repro de la liste V
    # et on attribue cette liste à Village pour la génération suivante
```

```
traitementresultats(RESULTATS)
```

```
miseenfichiercsv(lista1,lista2,lista3,GEN)
```