

### 3 L'expérience de Luria–Delbrück

#### 3.1 Exercices

**Exercice 16 :** On réalise l'expérience de Luria–Delbrück pour 30 populations de bactéries, chacune de taille finale  $n = 10000$ . Après contact avec les phages, voici les nombres de bactéries survivantes dans chacune des trente colonies :

10, 31, 20, 27, 186, 179, 21, 25, 15, 57, 28, 41, 33, 34, 94,  
190, 34, 12, 32, 21, 14, 37, 70, 34, 15, 49, 123, 45, 35, 19.

- Calculer la moyenne et la variance de l'échantillon. Ces données semblent-elles compatibles avec l'hypothèse  $H_0$  d'immunité acquise ?
- Mettre en place un test statistique permettant de tester  $H_0$ . Peut-on accepter  $H_0$  à un niveau de confiance de 99% ?

#### 3.2 Programmation

**Exercice 17 :** Le but de cet exercice est de simuler l'expérience de Luria–Delbrück.

- Écrire une fonction  $LD(n, p_1)$  qui simule un arbre de Yule arrêté à  $n$  individus, où  $p_1$  désigne la probabilité de mutation à la naissance.  
L'information portée par un nœud de l'arbre est un booléen indiquant si oui ou non l'individu correspondant porte la mutation de résistance aux phages – cette résistance est bien sûr héréditaire.
- Écrire une fonction  $alive(tree)$  qui étant donné un arbre renvoyé par la fonction ci-dessus, compte le nombre d'individus survivants à la mise en contact avec les phages.
- Générer un échantillon simulé sur lequel on testera l'hypothèse  $H_0$ .

**Exercice 18 :**

- En utilisant le code ci-dessus pour  $n = 1000$ ,  $p_1 = 0.005$  et en effectuant un grand nombre de simulations (nombre à choisir judicieusement, ne pas faire plus de 1000 répétitions), donner une estimation du quantile 99% de la distribution de la variable aléatoire  $\bar{X}/\sigma^2(X)$ , où  $X$  est un échantillon de 30 réalisations de l'expérience.
- Tester les données  
`sample = [16, 7, 5, 34, 5, 30, 7, 10, 29, 11, 15, 26, 151, 24, 21,  
17, 19, 5, 6, 6, 18, 12, 14, 246, 9, 33, 16, 16, 5, 23]`  
contre l'hypothèse  $H_1$ .
- En utilisant la fonction `numpy.random.poisson()` (d'abord penser à *importer* `numpy`), générer un échantillon de longueur 30 selon l'hypothèse  $H_0$  avec un paramètre  $\lambda$  judicieux ( $\lambda = np_1 \log(n)$ ), et le tester contre  $H_1$ .  
Répéter le test mille fois pour estimer la puissance, sous l'hypothèse  $H_0$ , du test contre  $H_1$ .