Chaînes de Markov avancées

M2 Génie Biologique et Informatique – Premier semestre 2012–2013

Mikael Falconnet
mikael.falconnet@genopole.cnrs.fr

Table des matières

L'objectif de ce cours est de familiariser les étudiants avec les chaînes de Markov à temps continu et en particulier à la modélisation de l'évolution de séquences biologiques.

On introduira la loi exponentielle et les processus de Poisson, la notion de générateur infinitésimal sera abordée à travers les modèles markoviens de substitutions de nucléotides et nous évoquerons également les modèles de substitution de codons.

La simulation sous le logiciel `R` occupera une part importante du module.

1 Processus de Poisson

1.1 Rappels sur la loi exponentielle

La loi exponentielle joue un rôle fondamental dans les processus de Poisson et les chaînes de Markov à temps continu. Nous en exposons ici les propriétés principales.

Définition 1.1 (loi exponentielle)**.** *La loi exponentielle de paramètre λ, (avec $\lambda > 0$), notée* $\mathrm{Exp}(\lambda)$ *est la loi de densité*

$$x \mapsto \lambda \mathrm{e}^{-\lambda x}, \quad x \geqslant 0.$$

En particulier, C suit une loi exponentielle de paramètre λ si et seulement si

$$\mathbb{P}(C > t) = \mathrm{e}^{-\lambda t}, \quad \textit{pour tout } t \geqslant 0.$$

La notation C est un choix personnel et provient de la première lettre de *clock* signifiant *horloge* en anglais. En effet, dans la majorité des situations rencontrées, les variables aléatoires suivant des lois exponentielles représenteront des temps d'attente entre deux arrivées. On peut donc se représenter C comme le temps de sonnerie d'une horloge aléatoire suivant une loi exponentielle. La notation T sera réservée aux temps d'arrivée.

Proposition 1.2. *La loi exponentielle satisfait les propriétés suivantes.*

(i) $\mathbb{P}(C > t + u \,|\, C > u) = \mathbb{P}(C > t) = \mathrm{e}^{-\lambda t}$, *pour tous* $t, u > 0$.

(ii) $\mathbb{P}(C \leqslant t + s \,|\, C > t) = \lambda s[1 + \varepsilon(s)]$, *où* $\varepsilon(s) \to 0$ *quand* $s \to 0$.

(iii) Soient C_1, C_2, …, C_n des variables aléatoires indépendantes distribuées suivant des lois exponentielles de paramètres λ_1, λ_2, …, λ_n.
Alors $C = \min(C_1, \ldots, C_n)$ suit une loi exponentielle de paramètre $\sum_{i=1}^{n} \lambda_i$.

La propriété (i), appelée *absence de mémoire*, caractérise la loi exponentielle, c'est à dire que c'est la seule loi continue qui possède cette propriété. La démonstration est un élégant exercice sur les équations fonctionnelles. Plus

important est l'interprétation de (i). Elle signifie que si l'horloge n'a pas sonné jusqu'au temps u, la probabilité qu'elle sonne dans l'intervalle $(u, t+u]$ ne dépend que de t et pas de u. L'horloge a "oublié" qu'elle n'avait pas sonné jusqu'au temps u et s'est réinitialisée. Cette propriété peut sembler perturbante quand on pense que C peut représenter le temps de panne d'une machine. En effet, l'absence de mémoire dans ce cas revient à oublier l'usure de la machine.

Le propriété (ii) signifie que si une horloge suit une loi exponentielle de paramètre λ, la probabilité qu'elle sonne dans un petit intervalle de longueur s est approximativement λs.

La propriété (iii) signifie que si je regarde n horloges indépendantes alors le temps d'attente de la première sonnerie parmi ces n horloges est aussi une loi exponentielle.

Preuve de la proposition 1.2. Montrons (i). Nous avons

$$\mathbb{P}(C > t+u \,|\, C > u) = \frac{\mathbb{P}(C > t+u)}{\mathbb{P}(C > u)} = \frac{\mathrm{e}^{-\lambda(t+u)}}{\mathrm{e}^{-\lambda u}} = \mathrm{e}^{-\lambda t}.$$

Montrons (ii). Nous avons

$$\mathbb{P}(C \leqslant t+s \,|\, C > t) = 1 - \mathbb{P}(C > t+s \,|\, C > t) = 1 - \mathrm{e}^{-\lambda s} = \lambda s[1+\varepsilon(s)].$$

Montrons (iii). Nous avons par définition de C

$$\mathbb{P}(C > t) = \mathbb{P}(C_1 > t, \dots C_n > t),$$

puis grâce à l'indépendance des variables aléatoires C_1, C_2, …, C_n

$$\mathbb{P}(C > t) = \prod_{i=1}^{n} \mathbb{P}(C_i > t) = \prod_{i=1}^{n} \mathrm{e}^{-\lambda_i t} = \mathrm{e}^{-t\sum_{i=1}^{n}\lambda_i}. \qquad \square$$

Exercice I (Illustration de la propriété (iii) de la proposition 1.2)**.**

1. *Rappeler la fonction de densité d'une loi exponentielle de paramètre $\lambda = 2$ et tracer sa courbe à l'aide de* `R`*. On pourra utiliser les fonctions* `plot` *et* `dexp`*.*
2. *Simuler un n-échantillon d'une loi exponentielle de paramètre $\mu > 0$ avec $n = 500$ et $\mu = 1$. On pourra utiliser la fonction* `rexp`*.*
3. *Simuler un n-échantillon du minimum de deux lois exponentielles indépendantes de paramètres $\mu_1 = \mu_2 = 1$ avec $n = 500$. On pourra utiliser la fonction* `pmin`*.*
4. *Tracer l'histogramme d'un n-échantillon du minimum de deux exponentielles indépendantes de paramètre $\mu_1 = \mu_2 = 1$, avec $n = 500$. On pourra utiliser la fonction* `hist`*.*

5. *Tracer sur le même graphique la fonction de densité d'une loi exponentielle de paramètre $\lambda = 2$ et l'histogramme d'un n-échantillon du minimum de deux exponentielles indépendantes de paramètre $\mu_1 = \mu_2 = 1$ avec $n = 500$. On pourra utiliser l'option* `breaks` *dans la fonction* `hist` *pour affiner l'histogramme.*
6. *Reprendre la question 4. avec $n = 1000$.*
7. *Êtes-vous convaincu quant à la propriété (iii) de la proposition 1.2 ?*

Plutôt que d'utiliser un histogramme pour représenter un n-échantillon et se demander si il correspond à la "bonne" loi de densité, nous allons utiliser un autre outil : la fonction de répartition empirique.

Définition. *Soit X une variable aléatoire réelle. La fonction de répartition F de X est donnée par*

$$F(x) = \mathbb{P}(X \leqslant x).$$

Définition. *En statistique, la fonction de répartition empirique $F_n(\cdot)$ d'un n-échantillon $X_1, \ldots, X_n$ est une fonction aléatoire en escalier définie par*

$$F_n(x) = \frac{\textit{nombre d'éléments} \leqslant x \textit{ dans l'échantillon}}{n} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{X_i \leqslant x\}.$$

Théorème (Théorème de Glivenko-Cantelli)**.** *Soit $X_1, \ldots, X_n$ un n-échantillon où chaque X_i a pour fonction de répartition F. Alors presque sûrement, la fonction de répartition empirique $F_n(\cdot)$ converge uniformément vers F quand n tend vers l'infini.*

Exercice II (Une meilleure illustration de la propriété (iii) de la proposition 1.2)**.**

1. *On suppose que $X_1 = 2$, $X_2 = 1$, $X_3 = 2.5$ et $X_4 = 3.5$. Tracer à la main puis à l'aide de* `R` *la fonction de répartition empirique de cet échantillon. On pourra utiliser la fonction* `ecdf`*. Dans un souci esthétique, on utilisera les options* `verticals=TRUE` *et* `do.points=FALSE`*.*
2. *Donner la fonction de répartition d'une loi exponentielle de paramètre $\lambda = 2$ et tracer sa courbe à l'aide de* `R`*.*
3. *Tracer sur le même graphique la fonction de répartition d'une loi exponentielle de paramètre $\lambda = 2$ et la fonction de répartition empirique d'un n-échantillon du minimum de deux exponentielles indépendantes de paramètre $\mu_1 = \mu_2 = 1$ avec $n = 500$.*
4. *Reprendre la question 3. avec $n = 1000$.*
5. *Êtes-vous convaincu quant à la propriété (iii) de la proposition 1.2 ?*

1.2 Processus de Poisson : définition et propriété principale

Parmi les processus stochastiques à temps continu, le processus de Poisson occupe une place privilégiée. Il est utilisé pour décrire la réalisation dans le temps d'évènements aléatoires d'un type donné. Classiquement, on retrouve l'arrivée de clients à un guichet, l'arrivée de demandes de tâches sur une imprimante, l'occurrence d'accidents dans une entreprise, etc. Schématiquement, ceci revient à modéliser les temps de sonnerie d'une horloge aléatoire.

Définition 1.3 (Processus de comptage)**.** *La description mathématique d'un flux d'évènements aléatoires peut se faire de deux manières différentes.*

1. *On considère le nombre d'évènements $N(t)$ se produisant dans l'intervalle de temps $[0,t]$ et on cherche à déterminer la distribution de cette variable aléatoire discrète. Le processus stochastique $\{N(t) : t \geqslant 0\}$ est appelé* processus de comptage*, ses réalisations sont des fonctions en escalier croissantes (figure 1). Notons que $N(u+t) - N(u)$ indique le nombre aléatoire d'évènements se produisant dans l'intervalle $]u, t+u]$.*
2. *On considère les intervalles de temps qui séparent les instants d'apparition de deux évènements consécutifs. Ce sont des variables aléatoires continues et positives dont on admettra généralement qu'elles sont indépendantes et identiquement distribuées. La connaissance de leur distribution commune permettra alors de déterminer les propriétés du processus de comptage correspondant.*

Soit $\{N(t) : t \geqslant 0\}$ un processus de comptage et désignons par C_n la durée séparant le $(n-1)$-ième évènement du n-ième pour $n \geqslant 1$. La variable aléatoire C_n, appelé *temps d'attente*, représente le temps pendant lequel le processus demeure dans l'état $n-1$. Posons ensuite

$$T_n = C_1 + C_2 + \cdots + C_n,$$

qui est le temps écoulé jusqu'à la réalisation du n-ième évènement. On vérifie aisément que

$$[N(t) \leqslant n] \Leftrightarrow [T_{n+1} > t] \quad \text{et} \quad [N(t) \geqslant n] \Leftrightarrow [T_n \leqslant t].$$

Définition 1.4 (Processus de Poisson)**.** *On dit qu'un processus de comptage $\{N(t) : t \geqslant 0\}$ est un* processus de Poisson *s'il satisfait aux quatre conditions suivantes.*

(H1) Le processus $N(t)$ est homogène dans le temps. Ceci veut dire que la probabilité d'avoir k évènements dans un intervalle de longueur donnée

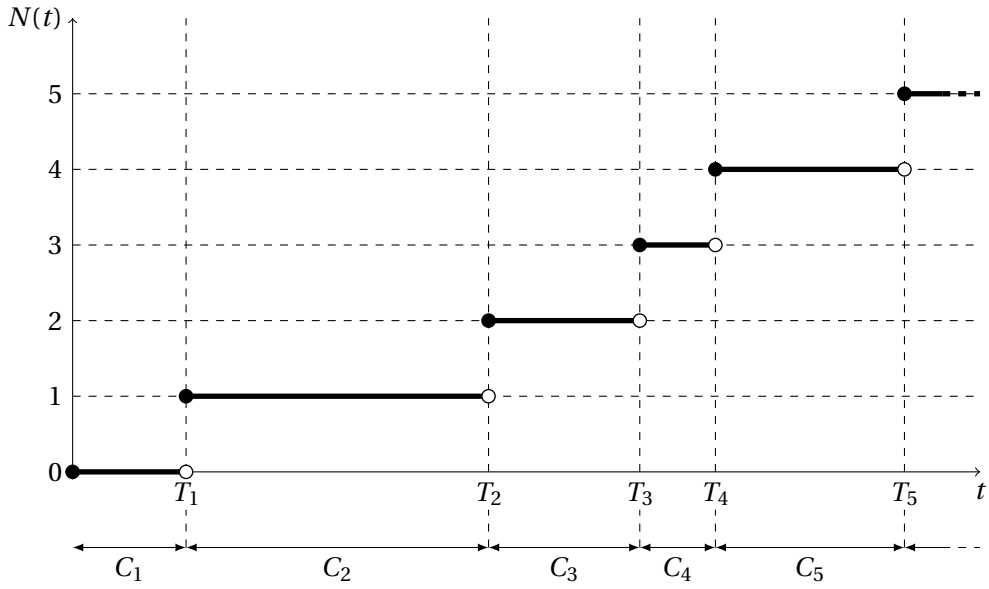


FIGURE 1 – Réalisation d'un processus de comptage

t ne dépend que de t et non pas de la position de l'intervalle par rapport à l'axe temporel. En d'autres termes

$$\mathbb{P}(N(s+t)-N(s)=k)=\mathbb{P}(N(t)=k)=p_k(t),$$

pour tous $s>0$, $t>0$ et $k=0,1,\ldots$.

(H2) Le processus $N(t)$ est à accroissements indépendants, ce qui signifie que pour tout système d'intervalles disjoints, les nombres d'évènements s'y produisant sont des variables aléatoires indépendantes. En particulier,

$$\mathbb{P}(N(t+s)-N(s)=k, N(s)=\ell)=p_k(t)p_\ell(s).$$

(H3) La probabilité qu'un évènement se produise dans un petit intervalle de temps est proportionnelle à la longueur de cet intervalle, le coefficient de proportionnalité étant λ. Autrement dit,

$$p_1(s)=\lambda s+s\varepsilon(s), \quad \text{où } \varepsilon(s)\to 0 \text{ quand } s\to 0.$$

Le coefficient λ est appelé intensité *du processus poissonien.*

(H4) La probabilité que deux évènements ou plus se produisent dans un petit intervalle s est négligeable par rapport à la probabilité qu'il n'y ait qu'un seul évènement. En termes plus précis

$$p_k(s)=s\varepsilon(s), \quad \text{où } \varepsilon(s)\to 0 \text{ quand } s\to 0 \text{ et } k\geqslant 2.$$

Cette condition exclut la possibilité d'une réalisation simultanée de deux évènements ou plus.

Théorème 1.5. *Si un processus de comptage satisfait aux conditions (H1)-(H4), alors*

$$\mathbb{P}(N(t) = k) = p_k(t) = \mathrm{e}^{-\lambda t}\frac{(\lambda t)^k}{k!},$$

et

$$\mathbb{E}[N(t)] = \lambda t, \quad \mathrm{Var}[N(t)] = \lambda t.$$

La variable aléatoire $N(t)$ suit une donc une loi de Poisson de paramètre λt. La démonstration de ce résultat est admise.

1.3 Processus de Poisson et loi exponentielle

Soit $\{N(t) : t \geqslant 0\}$ un processus de Poisson de paramètre λ et C_n la durée séparant le $(n-1)$-ième évènement du n-ième.

Théorème 1.6. *Les temps d'attente d'un processus de Poisson de paramètre λ sont des variables aléatoires indépendantes et identiquement distribuées selon une loi exponentielle de paramètre λ.*

Heuristique de la preuve. Considérons d'abord C_1, le temps d'attente du premier évènement. Nous avons

$$\mathbb{P}(C_1 > t) = \mathbb{P}(\text{aucun évènement dans } [0, t]) = \mathbb{P}(N(t) = 0) = \mathrm{e}^{-\lambda t}.$$

La variable aléatoire C_1 suit donc bien une loi exponentielle. Ensuite

$$\mathbb{P}(C_2 > t \,|\, C_1 = s) = \mathbb{P}(\text{aucun évènement dans }]s, s+t] \,|\, C_1 = s)$$

En utilisant la condition (H2), on peut supprimer le conditionnement et il vient

$$\mathbb{P}(C_2 > t \,|\, C_1 = s) = \mathbb{P}(\text{aucun évènement dans }]s, s+t]) = \mathrm{e}^{-\lambda t}.$$

Ainsi

$$\mathbb{P}(C_2 > t, C_1 > u) = \int_u^{+\infty} \mathbb{P}(C_2 > t \,|\, C_1 = s)\lambda \mathrm{e}^{-\lambda s}\mathrm{d}s = \mathrm{e}^{-\lambda t}\mathrm{e}^{-\lambda u}.$$

Comme annoncé, ceci est une heuristique. L'évènement $[C_1 = s]$ est de probabilité nulle puisque C_1 est une variable aléatoire continue, le conditionnement "classique" est donc prohibé et découle d'une théorie un peu plus poussée que les connaissances exigées pour ce cours. Pour comprendre l'égalité précédente, on peut faire une analogie avec le cas d'une variable aléatoire discrète. En effet, si C_1 est à valeurs entières, on a

$$\mathbb{P}(C_2 > t, C_1 > u) = \sum_{s>u} \mathbb{P}(C_2 > t \,|\, C_1 = s)\mathbb{P}(C_1 = s).$$

Dans le cas continu, on remplace la somme par une intégrale et $\mathbb{P}(C_1 = s)$ par la probabilité que C_1 se trouve dans un petit intervalle de longueur $\mathrm{d}s$ autour de s à savoir $\lambda \mathrm{e}^{-\lambda s}\mathrm{d}s$, où le premier terme représente la densité de C_1. □

Corollaire 1.7. *La durée séparant $n+1$ évènements, en d'autres termes le temps écoulé entre le k-ième et le $(k+n)$-ième évènement, obéit à une loi gamma de paramètres λ et n.*

Le théorème 1.6 admet une réciproque et permet de caractériser différemment un processus de Poisson

Théorème 1.8. *Soit $\{N(t) : t \geqslant 0\}$ un processus de comptage. Si les intervalles de temps entre deux évènements consécutifs sont des variables aléatoires indépendantes obéissant à la même loi exponentielle de paramètre λ, alors le processus est un processus de Poisson de paramètre λ.*

Exercice III.

Admettons qu'à la poste de Saint-Donat-sur-l'Herbasse, l'arrivée des clients suive un processus de Poisson de paramètre $\lambda = 5\mathrm{h}^{-1}$.

1. *Quelle est la probabilité d'avoir 11 clients ou plus dans la matinée de 9h30 à 11h30 ?*
2. *Entre 9h30 et 10h00, un seul client s'est présenté à la poste. Quelle est la loi de son temps d'arrivée sachant cette information ?*

Définition. *Soit $X_1, \ldots, X_n$ un n-échantillon. On appelle moyenne empirique du n-échantillon la quantité $\bar{X}_n$ définie par*

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{1}{n}(X_1 + \cdots + X_n).$$

Théorème (Loi des grands nombres)**.** *Quand n tend vers l'infini, la moyenne empirique $\bar{X}_n$ d'un n-échantillon de variables aléatoires indépendantes converge vers la moyenne théorique $\mathbb{E}(X_1)$.*

Exercice IV (Simuler un processus de Poisson avec un nombre de sauts fixé)**.**

1. *Écrire une fonction* `PoissonSaut` *qui prend en entrée une intensité $\lambda > 0$ et un entier $n \geqslant 1$ et qui renvoie les n premiers temps d'arrivée d'un processus de Poisson d'intensité λ. On pourra utiliser les fonctions* `rexp` *et* `cumsum`*.*
2. *Afficher la trajectoire d'un processus de Poisson d'intensité $\lambda = 2$ jusqu'au 10-ième saut. On pourra utiliser les fonctions* `PoissonSaut` *et* `stepfun`*.*
3. *Écrire une matrice comportant $n = 10$ lignes et $m = 1000$ colonnes telle que chaque colonne contienne les n premiers temps d'arrivée d'un processus de Poisson d'intensité $\lambda = 2$. On pourra utiliser les fonctions* `matrix` *et* `apply`*.*

4. *On note $T_1^{(i)}$ le temps d'arrivée du premier évènement dans le i-ème processus de Poisson, c'est à dire le premier terme de la i-ème colonne de la matrice générée précédemment. On définit $\bar{T}_1^{(i)}$ comme la moyenne empirique du i-échantillon $T_1^{(1)}, T_1^{(2)}, \ldots, T_1^{(i)}$. Afficher la suite $\{\bar{T}_1^{(i)} : i = 1 \ldots m\}$ et superposer à l'aide de* `lines` *la moyenne d'une loi exponentielle de paramètre $\lambda = 2$. Que constate-t-on ?*
5. *Reprendre la question 4. avec $m = 5000$.*
6. *Reprendre les questions 4. et 5. avec $T_5^{(i)}$ le temps d'arrivée du premier évènement dans le i-ème processus de Poisson. Quelle est la limite de la suite $\{\bar{T}_5^{(i)} : i = 1 \ldots m\}$?*

Exercice V (Simuler un processus de Poisson jusqu'à un temps fixé)**.**

1. *Écrire une fonction* `PoissonTemps` *qui prend en entrée une intensité $\lambda > 0$ et un temps $t \geqslant 0$ et qui renvoie les temps d'arrivée d'un processus de Poisson d'intensité λ jusqu'au temps t. On pourra utiliser une boucle* `while`*.*
2. *Simuler et afficher la trajectoire d'un processus de Poisson d'intensité $\lambda = 2$ jusqu'à l'instant $t = 10$.*
3. *Simuler la trajectoire d'un processus de Poisson d'intensité $\lambda = 2$ jusqu'à l'instant $t = 100$ et afficher les rapports $N(T_i)/T_i$ pour tous les $T_i < t$. Que constate-t-on ?*

1.4 Propriétés supplémentaires

Soient $\{N_1(t) : t \geqslant 0\}$ et $\{N_2(t) : t \geqslant 0\}$ deux processus de Poisson indépendants de paramètres λ_1 et λ_2. Ceci signifie que tout évènement défini par le premier processus est indépendant de tout évènement défini par le second.

Théorème 1.9 (Superposition)**.** *Le processus $\{N(t) = N_1(t) + N_2(t) : t \geqslant 0\}$ est un processus de Poisson de paramètre $\lambda_1 + \lambda_2$.*

Le théorème 1.9 peut s'interpréter de la manière suivante. Si à la poste de Saint-Donat-sur-l'Herbasse, l'arrivée des clients suit un processus de Poisson de paramètre $5\mathrm{h}^{-1}$ et si à la poste de Plougastel-Daoulas, elle suit un processus de Poisson de paramètre $6\mathrm{h}^{-1}$ et si ces processus sont indépendants (ce qui compte tenu de la distance paraît raisonnable) alors l'arrivée des clients dans une des deux postes suit une loi de Poisson de paramètre $11\mathrm{h}^{-1}$.

Considérons maintenant la situation inverse. Un processus de Poisson $\{N(t) : t \geqslant 0\}$ est décomposé en deux processus partiels de la façon suivante. Chaque évènement est soumis à une expérience de Bernoulli qui l'attribue à l'un des deux types 1 ou 2 avec des probabilités p et $1 - p$. On admet que ces attributions sont indépendantes les unes des autres et indépendantes de l'état

du processus $N(t)$. Notons $\{N_i(t) : t \geqslant 0\}$ le processus formé des évènements du type i $(i = 1, 2)$.

Théorème 1.10 (Décomposition)**.** *Les processus N_1 et N_2 sont deux processus de Poisson indépendants de paramètres λp et $\lambda(1-p)$.*

Voici un théorème important en terme de simulation de processus de Poisson.

Théorème 1.11 (Loi conditionnelle des temps de saut)**.** *Soit $\{N(t) : t \geqslant 0\}$ un processus de Poisson de paramètre λ. Sachant que $N(t) = n$, avec $n \geqslant 1$, la loi du n-uplet $(T_1, \ldots, T_n)$ est la même que celle d'un n-échantillon de variables aléatoires i.i.d. de loi uniforme sur $[0, t]$.*

Exercice VI (une meilleure manière de simuler un processus de Poisson jusqu'à un temps fixé)**.**

Soit $\{N(t) : t \geqslant 0\}$ un processus de Poisson d'intensité λ.

1. *Rappeler la loi de $N(t)$ quand t est fixé. Écrire une fonction* `NombreSaut` *qui prend en entrée un temps $t \geqslant 0$ et une intensité $\lambda > 0$ et qui renvoie le nombre d'évènements qui se sont produits dans un processus de Poisson d'intensité λ jusqu'au temps t. On pourra utiliser* `rpois`*.*
2. *En utilisant le théorème 1.11, écrire une fonction* `PoissonTemps2` *qui prend en entrée un temps $t \geqslant 0$ et une intensité $\lambda > 0$ et qui renvoie les temps d'arrivée d'un processus de Poisson d'intensité λ jusqu'au temps t sans utiliser de boucle* `while`*. On pourra utiliser les fonctions* `NombreSaut`*,* `runif` *et* `sort`*.*
3. *Comparer les temps de calculs des fonctions* `PoissonTemps` *et* `PoissonTemps2` *pour $t = 10000$ et $\lambda = 1$. On utilisera la fonction* `system.time`*.*
4. *Quelle est votre conclusion ?*

2 Modèles d'évolution de séquences biologiques

2.1 Modèles markoviens de substitution de nucléotides

Dans une séquence d'ADN (réciproquement d'acides aminés), une substitution est le remplacement d'un nucléotide (réciproquement d'un acide aminé) par un autre. Ce phénomène est une source importante de mutation pour les séquences biologiques et sa modélisation peut être mise en place grâce aux chaînes de Markov à temps continu.

Plutôt que de développer une théorie sur cette classe de processus, nous allons nous concentrer sur des modèles concrets en commençant par le modèle de Jukes et Cantor.

Le modèle de Jukes et Cantor est une chaîne de Markov à temps continu et c'est le modèle de substitution nucléotidique le plus élémentaire. Fixons une fois pour toute notre cadre de travail.

Définition 2.1. *L'alphabet des nucléotides est*

$$\mathscr{A} = \{\mathtt{a}, \mathtt{t}, \mathtt{c}, \mathtt{g}\}.$$

Ce sont les premières lettres respectives pour adénine, thymine, cytosine et guanine. Les nucléotides `a` *et* `g` *sont des purines représentées par* `r`*, les nucléotides* `t` *et* `c` *des pyrimidines, représentées par* `y`*.*

Une séquence d'ADN de longueur N est donc représenté par un élément de $\mathscr{A}^N$. On oublie donc la structure en double-hélice et on se concentre sur un seul brin d'ADN.

Dans le modèle de Jukes et Cantor, on va supposer que chaque site de la séquence évolue *indépendamment* des autres sites. Ainsi, l'évolution d'une séquence d'ADN de longueur N sous ce modèle n'est rien de plus que l'évolution parallèle des N sites qui la composent. Une autre hypothèse importante et caractéristique du modèle est l'homogénéité des substitutions qui ont toutes lieu au même taux.

Définition 2.2. *Soit $X_i(t)$ la variable aléatoire représentant la nature du nucléotide occupant le site $i \in \{1, \dots, N\}$ au temps $t \geqslant 0$.*

Dans le modèle JC, le processus $\{X_i(t) : t \geqslant 0\}$ est un processus de Markov (ou chaîne de Markov à temps continu) à valeurs dans $\mathscr{A}$, et il existe un paramètre $\lambda > 0$ tel que le générateur $Q = \{q(x,y) : (x,y) \in \mathscr{A}^2\}$ soit donné par la matrice 4×4 de taux de substitutions

$$Q = \begin{array}{c} \\ \mathtt{a} \\ \mathtt{t} \\ \mathtt{c} \\ \mathtt{g} \end{array} \begin{array}{c} \begin{array}{cccc} \mathtt{a} & \mathtt{t} & \mathtt{c} & \mathtt{g} \end{array} \\ \begin{pmatrix} -3\lambda & \lambda & \lambda & \lambda \\ \lambda & -3\lambda & \lambda & \lambda \\ \lambda & \lambda & -3\lambda & \lambda \\ \lambda & \lambda & \lambda & -3\lambda \end{pmatrix} \end{array}. \tag{2.1}$$

Il y a plusieurs termes à comprendre dans la définition 2.2. Nous n'insisterons par sur ce qu'est une chaîne de Markov à temps continu et ses multiples définitions possibles. Pour faire simple, nous dirons que ce n'est rien de plus que la généralisation d'un processus de Poisson, à savoir un processus sans mémoire avec des temps de sauts aléatoires séparés par des horloges suivant des lois exponentielles indépendantes, le tout couplé avec des sauts qui ne sont plus déterministes ($+1$ dans le cas d'un processus de Poisson) mais aléatoires. Toutes ces informations sont contenues dans le générateur Q que nous décrivons maintenant.

Chaque terme hors diagonal $q(x,y)$ représente le taux de substitution du nucléotide x par le nucléotide y. Cela signifie que pendant un petit intervalle de longueur s, la probabilité que x soit remplacé par y est donnée par $q(x,y)s$. Ceci n'est pas sans rappeler la propriété (ii) de la proposition 1.2 et la condition (H3) de la définition 1.4.

Chaque terme $-q(x,x)$ peut être interprété comme le taux de substitution du nucléotide x. Concrètement, cela signifie que l'on associe une horloge aléatoire suivant une loi exponentielle de paramètre $-q(x,x)$ au site i occupé par le nucléotide x et quand l'horloge sonne on remplace le nucléotide x par le nucléotide $y \in \mathscr{A} \setminus \{x\}$ avec probabilité $-q(x,y)/q(x,x)$.

Dans le modèle JC, le nucléotide `c` est remplacé par le nucléotide `a`, `t` ou `g` avec probabilité $1/3$ et le taux de substitution de `c` est de 3λ. Le paramètre λ représente donc un taux de mutation, plus il est important, plus le nombre de substitutions moyen dans un intervalle de temps est élevé

Exercice VII (Simulation de l'évolution d'un site dans le modèle de Jukes et Cantor)**.**

On considère le modèle de Jukes et Cantor de paramètre $\lambda > 0$ où le générateur Q est défini par (2.1)*.*

1. *Donner pour chaque nucléotide le paramètre de l'horloge exponentielle qui lui est associée. Que constate-t-on ?*
2. *Écrire une fonction* `TempsSubstitutionJC` *qui prend en entrée un temps $t \geqslant 0$ et un paramètre $\lambda > 0$ et qui renvoie les temps de substitutions qui se produisent en un site sous le modèle de Jukes et Cantor ainsi que leur nombre. On évitera l'emploi d'une boucle* `while`*.*

Afin de faciliter la programmation, plutôt que d'utiliser l'alphabet $\mathscr{A}$, on utilise l'ensemble $\{1,2,3,4\}$ pour coder les nucléotides. Ainsi, 1 correspond à `a`*, 2 correspond à* `t`*, 3 correspond à* `c` *et 4 correspond à* `g`*.*

3. *À l'aide de la fonction* `sample`*, écrire une fonction* `SubstitutionJC` *qui prend en entrée un entier $i \in \{1,2,3,4\}$ et renvoie un entier choisi uniformément parmi $\{1,2,3,4\} \setminus \{i\}$.*
4. *Écrire une fonction* `EvolutionJC` *qui prend en entrée un temps $t \geqslant 0$ et un paramètre $\lambda > 0$ et qui renvoie une matrice à deux lignes contenant dans la première ligne les temps de substitutions en un site sous le modèle de Jukes et Cantor de paramètre λ jusqu'au temps t et dans la deuxième ligne les substitutions associées. On utilisera une boucle* `for`*.*
5. *Simuler et afficher l'évolution d'un site dans l'ensemble $\{1,2,3,4\}$ sous le modèle de Jukes et Cantor de paramètre $\lambda = 1$ jusqu'au temps $t = 10$.*

Dans un souci esthétique, nous allons convertir les valeurs entières en couleurs et représenter l'évolution d'un site sous forme d'une bande de couleurs.

6. *Utiliser la fonction* `rect` *pour afficher l'évolution d'un site sous le modèle de Jukes et Cantor de paramètre* $\lambda = 1$ *jusqu'au temps* $t = 10$*. On utilisera la couleur correspondant à l'adenine est le rouge, à la thymine le bleu, à la cytosine le vert et à la guanine le jaune. On examinera la fonction* `rect` *et on modifiera la fonction* `TempsSubstitutionJC` *en conséquence.*

Dans l'exercice VII, pour simuler l'évolution d'un site, on a recours à une boucle `for` pour choisir à chaque instant de substitution le nouveau nucléotide qui va remplacer l'ancien. Pour éviter ce désagréement, nous allons utiliser une modélisation alternative du modèle de Jukes et Cantor.

Définition 2.3. *Soient* $\{N_x(t) : t \geqslant 0\}$ *avec* $x \in \mathscr{A}$*, quatre processus de Poisson indépendants d'intensité* $\lambda > 0$*. On marque chaque évènement associé au processus* $N_x(\cdot)$ *par* U_x*. On définit le processus* $X_i(\cdot)$ *de la manière suivante :* $X_i(0)$ *est la valeur initiale du processus, puis au cours du temps à chaque fois qu'il rencontre une marque* U_x *en* t*, il se transforme en* x*. Bien sûr, si* $X_i(t^-) = x$*, rien ne se passe.*

Proposition 2.4. *Le processus* $X_i(\cdot)$ *défini en 2.3 suit un modèle de Jukes et Cantor de paramètre* $\lambda > 0$*.*

Démonstration. Nous allons uniquement vérifier que le processus a les bonnes transitions $q(x,y)$ définis en (2.1). Soient $s > 0$ petit et $t > 0$ fixé. Comme les processus $\{N_x(t) : t \geqslant 0\}$ sont indépendants, la probabilité d'avoir deux marques ou plus dans un intervalle de longueur s est de l'ordre de s^2. Ainsi, nous avons

$$\mathbb{P}(X_i(t+s) = y | X_i(t) = x) = \mathbb{P}(N_y(t+s) - N_y(t) = 1, N_z(t+s) - N(t) = 0, z \neq y) + s\varepsilon(s).$$

Grâce à l'indépendance mutuelle des $N_x(\cdot)$, il vient

$$\begin{aligned}\mathbb{P}(X_i(t+s) = y | X_i(t) = x) = \mathbb{P}(N_y(t+s) - N_y(t) = 1) \\ \times \prod_{z \neq y} \mathbb{P}(N_z(t+s) - N(t) = 0, z \neq x) + s\varepsilon(s).\end{aligned}$$

Ainsi

$$\mathbb{P}(X_i(t+s) = y | X_i(t) = x) = \mathrm{e}^{-4\lambda s}\lambda s + s\varepsilon(s) = \lambda s + \tilde{\varepsilon}(s) = q(x,y)s + s\tilde{\varepsilon}(s). \quad \square$$

Exercice VIII.

1. *Écrire une fonction* `MarquageJC` *qui prend en entrée un temps* $t \geqslant 0$*, une intensité* λ *et un entier* $i \in \{1,2,3,4\}$ *et qui renvoie une matrice à deux lignes contenant dans la première ligne les temps d'arrivées d'un processus de Poisson de paramètre* λ *jusqu'au temps* t *et dans la deuxième ligne l'entier* i*.*

2. *Écrire une fonction* `EvolutionJC2` *qui prend en entrée un temps* $t \geqslant 0$ *et une intensité* λ *et qui renvoie une matrice dont la première ligne contient les temps d'arrivées rangés dans l'ordre croissant de quatre processus de Poisson indépendants et dans la seconde ligne les marques des processus associées. On pourra utiliser* `cbind` *et* `order`*.*
3. *Comparer les temps de calcul des fonctions* `EvolutionJC` *et* `EvolutionJC2`*.*

Le générateur infinitésimal Q détermine complètement la dynamique de la chaîne de Markov. Notons $p_{xy}(t)$ la probabilité que le site i soit occupé par y au temps t sachant qu'il est occupé par un x au temps 0, c'est à dire

$$p_{xy}(t) = \mathbb{P}(X_i(t) = y | X_i(0) = x).$$

Alors, la matrice de transition au temps t, notée $P(t) = (p_{xy}(t))$ est l'unique solution du problème de Cauchy

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = QP(t), \quad P(0) = \text{Identité},$$

et cette solution est donnée par

$$P(t) = \mathrm{e}^{Qt}.$$

Donnons la matrice de transition pour JC.

Proposition 2.5. *Dans JC, pour tous* x *et* $y \neq x$ *dans* $\mathscr{A}$*,* $p_{xy}(t) = \frac{1}{3}p(t)$ *et* $p_{xx}(t) = 1 - p(t)$*, avec*

$$p(t) = \frac{3}{4}\left(1 - \mathrm{e}^{-4\lambda t}\right).$$

Démonstration. Soient I la matrice identité 4×4 et J la matrice 4×4 dont tous les coefficients sont égaux à $\frac{1}{4}$. Alors $IJ = JI = J^2 = J$ et $Q = 4\lambda(J - I)$. Ainsi,

$$\mathrm{e}^{Qt} = \mathrm{e}^{-4\lambda t I}\mathrm{e}^{4\lambda t J} = \mathrm{e}^{-4\lambda t}\mathrm{e}^{4\lambda t J}.$$

Pour tout entier $n \geq 1$, $J^n = J$, ainsi

$$\mathrm{e}^{4\lambda t J} = \sum_{n \geqslant 0} \frac{(4\lambda t)^n}{n!} J^n = I + (\mathrm{e}^{4\lambda t} - 1)J.$$

Cela signifie que $\mathrm{e}^{Qt} = \mathrm{e}^{-4\lambda t} I + (1 - \mathrm{e}^{-4\lambda t})J$, ce qui achève la preuve. □

La donnée de la matrice de transition et de la mesure initiale de $X_i(0)$ caractérise complètement la loi de $X_i(t)$. En effet, supposons que $X_i(0)$ suive la loi initiale $\pi = (\pi_x)_{x \in \mathscr{A}}$, c'est à dire

$$\mathbb{P}(X_i(0) = x) = \pi_x,$$

alors la loi de $X_i(t)$, notée $\pi(t)$, est donnée par $\pi(t) = \pi P(t)$, c'est à dire que pour tout nucléotide x

$$\pi_x(t) = \mathbb{P}(X_i(t) = x) = \sum_{y \in \mathscr{A}} \pi_y p_{yx}(t).$$

Jusqu'à présent nous nous sommes concentrés sur l'évolution d'un seul site. Notons $X_{1:N}(t)$ le vecteur aléatoire $(X_1(t), X_2(t), \ldots, X_N(t))$ qui représente l'état d'une séquence d'ADN de longueur N au temps t.

Définition 2.6. *Dans le modèle de Jukes et Cantor, le processus $X_{1:N}(t)$ est markovien et si l'on suppose que $X_{1:N}(0) = x_{1:N}$, alors pour tous $y_{1:N} = (y_1, y_2, \ldots y_N) \in \mathscr{A}^N$*

$$\mathbb{P}^{x_{1:N}}(X_{1:N}(t) = y_{1:N}) = \prod_{i=1}^{N} \mathbb{P}^{x_i}(X_i(t) = y_i) = \prod_{i=1}^{N} \mathrm{e}^{Qt}_{x_i y_i}.$$

La définition 2.6 est simplement une traduction de l'hypothèse d'indépendance des N sites et de leur distribution identique.

La mesure de probabilité π^* sur $\mathscr{A}$ définie par $\pi^*_x = 1/4$ pour tout nucléotide x est invairante pour le modèle de Jukes et Cantor. Ceci signifie que si la mesure initiale de $X_i(0)$ est π^*, alors pour tout temps t, la loi de $X_i(t)$ est aussi π^*. De plus, le processus de Jukes et Cantor est *ergodique*, ce qui signifie que la loi de $X_i(t)$ converge vers π^* quand t tend vers l'infini. et ce quelle que soit la mesure initiale de $X_i(0)$.

Quand la longueur N de la séquence est grande, le nombre π^*_x peut s'interpréter comme la proportion de nucléotides x présents dans la séquence quand un temps assez long s'est écoulé. Ceci est une conséquence de la loi des grands nombres

$$\frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{X_i(t) = x\} \xrightarrow[N \to +\infty]{a.s.} \mathbb{E}(\mathbf{1}\{X_1(t) = x\}) = \pi_x(t) \approx \pi^*_x.$$

Exercice IX (Modèle de Kimura)**.**

Dans le modèle de Kimura, le processus $\{X_i(t) : t \geqslant 0\}$ est un processus de Markov (ou chaîne de Markov à temps continu) à valeurs dans $\mathscr{A}$, et il existe deux paramètres $w > v > 0$ tel que le générateur $Q = \{q(x, y) : (x, y) \in \mathscr{A}^2\}$ soit donné par la matrice 4×4 de taux de substitutions

$$Q = \begin{array}{c} \\ \mathtt{a} \\ \mathtt{t} \\ \mathtt{c} \\ \mathtt{g} \end{array} \begin{array}{c} \begin{array}{cccc} \mathtt{a} & \mathtt{t} & \mathtt{c} & \mathtt{g} \end{array} \\ \begin{pmatrix} \cdot & v & v & w \\ v & \cdot & w & v \\ v & w & \cdot & v \\ w & v & v & \cdot \end{pmatrix} \end{array}. \tag{2.2}$$

1. *Donner les termes diagonaux de Q et en déduire pour chaque nucléotide le paramètre de l'horloge exponentielle qui lui est associée. Que constate-t-on ?*
2. *Écrire une fonction* `TempsSubstitutionKimu` *qui prend en entrée un temps $t \geqslant 0$, un taux de transversion v et un taux de transition w, et qui renvoie les temps de substitutions qui se produisent en un site sous le modèle de Kimura. Si aucun évènement ne se produit, on renverra un vecteur vide. On évitera l'emploi d'une boucle* `while`.
3. *Sachant que le site* `i` *est occupé par un* `a` *et que l'horloge qui lui est associée sonne, donner les probabilités qu'il soit remplacé par un* `t`*, un* `c` *ou un* `g`.
4. *Reprendre la question 3. en remplaçant* `a` *par chacun des trois autres nucléotides.*
5. *Écrire une fonction* `SubstitutionKimu` *qui prend en entrée un taux de transversion v, un taux de transition w et un entier $i \in \{1,2,3,4\}$, et qui renvoie un entier $j \in \{1,2,3,4\} \setminus \{i\}$ choisi suivant les lois de probabilité établies en 3. et 4.*
6. *Écrire une fonction* `EvolutionKimu` *qui prend en entrée un temps $t \geqslant 0$, un taux de transversion $v > 0$, un taux de transition $w > v$ et un nucléotide initiale* `nucleo`*, et qui renvoie une matrice à deux lignes contenant dans la première ligne les temps de substitutions en un site sous le modèle de Kimura de paramètres v et w jusqu'au temps t et dans la deuxième ligne les substitutions associées.*

Comme dans le cas du modèle de Jukes et Cantor, une modélisation alternative est possible.

Définition. *Soient $\{M_x(t) : t \geqslant 0\}$ avec $x \in \mathscr{A}$, quatre processus de Poisson indépendants d'intensité $v > 0$. On marque chaque évènement associé au processus $M_x(\cdot)$ par V_x. Soient $\{N_x(t) : t \geqslant 0\}$ avec $x \in \mathscr{A}$, quatre processus de Poisson indépendants d'intensité $w - v > 0$. On marque chaque évènement associé au processus $N_x(\cdot)$ par W_x.On définit le processus $X_i(\cdot)$ de la manière suivante : $X_i(0)$ est la valeur initiale du processus, puis au cours du temps à chaque fois qu'il rencontre une marque V_x en t, il se transforme en x inconditionnellement. Bien sûr, si $X_i(t^-) = x$, rien ne se passe.En revanche, à chaque fois qu'il rencontre une marque W_x en t, il se transforme en x si c'est une transition qui a lieu, sinon il ne se passe rien.*

Proposition. *Le processus $X_i(\cdot)$ défini précédemment suit un modèle de Kimura de paramètres v et w.*

Exercice X (Modèle de Kimura, suite)**.**

1. *Démontrer la proposition en s'inspirant de ce qui a été fait en cours.*

2. *En s'inspirant de l'exercice VIII, écrire une nouvelle fonction* `EvolutionKimu2` *pour simuler l'évolution d'un site sous un modèle de Kimura.*
3. *Comparer les temps de calcul des fonctions* `EvolutionKimu` *et* `EvolutionKimu2` *.*
4. *Écrire une fonction* `EvolutionSite` *qui prend en entrée une séquence initiale, un temps $t \geqslant 0$, un taux de transversion $v > 0$ et un taux de transition $w > v$ et une longueur N, et qui renvoie la nature du nucléotide après une évolution sous le modèle de Kimura durant un temps t.*
5. *Écrire une fonction* `EvolutionSequence` *qui prend en entrée une séquence initiale, un temps $t \geqslant 0$, un taux de transversion $v > 0$ et un taux de transition $w > v$ et une longueur N, et qui renvoie une séquence ayant évolué sous le modèle de Kimura.*
6. *On prend comme séquence initiale une séquence de longueur $N = 10000$ ne comportant que des adénines. On fixe comme paramètres $v = 0.25$ et $w = 0.5$. Calculer les proportions empiriques de nucléotides pour la séquence ayant évolué pendant les temps $t = 0.1, 0.2, \ldots, 1$.*

2.2 Distance phylogénétique

Maintenant que nous avons introduit le modèle de Jukes et Cantor, nous allons voir comment l'utiliser dans un cadre phylogénétique, notamment pour calculer des distances phylogénétiques entre séquences d'ADN.

Considérons deux séquences d'ADN de longueur N. Supposons qu'une des séquences soit l'ancêtre de l'autre et que l'évolution de la séquence ait suivi un modèle de Jukes et Cantor durant un temps t inconnu. une telle hypothèse signifie que chaque site i de la séquence ancestrale est l'ancêtre du site i dans la séquence actuelle. Une telle hypothèse suppose que l'alignement des séquences est déjà effectué et nous n'évoquerons pas ce point. On cherche à retrouver à partir de l'alignement des deux séquences le temps t qui s'est écoulé.

Avant de se lancer dans la procédure, il faut regarder ce qu'on peut vraiment espérer retrouver. En effet, on peut remarquer que la loi de $X_i(t)$ dans un modèle de Jukes et Cantor avec paramètre λ est la même que celle de $X_i(t/2)$ dans un modèle de Jukes et Cantor avec paramètre 2λ. Ceci vient du fait que ces quantités apparaissent uniquement sous la forme du produit $d = \lambda t$ dans la matrice de transition $P(t)$. Ainsi, sans information extérieure sur le paramètre λ, il n'est pas possible d'estimer t en fonction de λ.

Dans le modèle de Jukes et Cantor, l'estimateur utilisé pour le temps écoulé est basé sur la proportion de sites différents entre les deux séquences. C'est

une application directe du maximum de vraisemblance, une méthode générale pour estimer des paramètres dans un modèle. Le principe du maximum de vraisemblance est le suivant : on suppose qu'on observe un jeu de données générées par un modèle probabiliste dépendant de paramètres et on regarde le jeu de paramètres qui rend le jeu de données le plus vraisemblable parmi tous les jeux de paramètres possibles.

Supposons par exemple que l'on dispose d'un échantillon $x_1, x_2, \dots, x_n$ de n observations indépendantes provenant d'une densité de probabilité f_0. Supposons de plus que la fonction f_0 appartienne à une certaine classe de fonctions de densité $\{f(\cdot|\theta) : \theta \in \Theta\}$ dépendante d'un paramètre θ, et que f_0 correspond au paramètre $\theta = \theta_0$ appelé vraie valeur du paramètre. L'idée qui se cache derrière la méthode du maximum de vraisemblance est de regarder la densité jointe $f(x_1, x_2, \dots, x_n|\theta)$ sous un angle différent. Considérons les observations $x_1, \dots, x_n$ comme des "paramètres" fixés de la fonction, et au contraire θ comme une variable. On construit ainsi une fonction de vraisemblance

$$L(\theta; x_1, \dots, x_n) = f(x_1, x_2, \dots, x_n|\theta).$$

En pratique, il est plus commode de travailler avec le logarithme de la vraisemblance, appelée log-vraisemblance

$$\ell(\theta; x_1, \dots, x_n) = \log L(\theta|x_1, \dots, x_n).$$

La méthode du maximum de vraisemblance consiste à approcher θ_0 par la valeur $\hat{\theta}$ qui maximise $\theta \mapsto \ell(\theta; x)$.

Dans le modèle de Jukes et Cantor, le seul paramètre est la distance d et les données sont le nombre de sites différents x entre les deux séquences. D'après la proposition 2.5, la probabilité d'observer deux sites différents est donnée par $p(t)$. Ainsi, la fonction de vraisemblance s'écrit

$$L(d; x) = \binom{N}{x} \left[\frac{3}{4}(1 - \mathrm{e}^{-4d})\right]^x \left[\frac{1}{4} + \frac{3}{4}\mathrm{e}^{-4d}\right]^{N-x},$$

et la log-vraisemblance est donnée (à une constante additive ne dépendant pas de d près) par

$$\ell(d; x) = x \log\left[\frac{3}{4}(1 - \mathrm{e}^{-4d})\right] + (N - x) \log\left[\frac{1}{4} + \frac{3}{4}\mathrm{e}^{-4d}\right].$$

On a

$$\frac{\mathrm{d}}{\mathrm{d}d}\ell(d; x) = 0 \Leftrightarrow p(t) = \frac{x}{N},$$

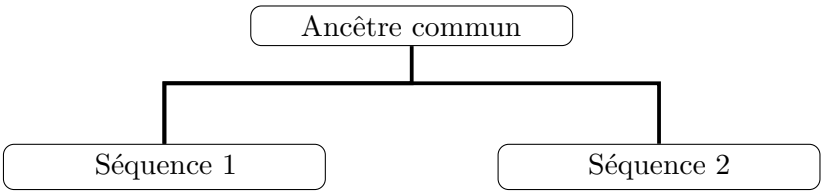
ce qui nous donne

$$\hat{d} = -\frac{1}{4}\log\left(1 - \frac{4}{3}p\right), \tag{2.3}$$

avec $p = x/N$ la proportion de sites différents entre les deux séquences.

L'estimateur du temps écoulé donné par (2.3) vérifie des propriétés de consistance et de normalité asymptotique que nous ne donnons pas dans ce cours.

Pour le moment, nous avons uniquement fourni un estimateur du temps écoulé entre une séquence ancestrale et une séquence actuelle. Rappelons que dans la majorité des cas, l'accès à la séquence fossile n'est pas possible et le problème que l'on veut résoudre est l'estimation du temps de divergence entre deux séquences actuelles issues d'une même séquence ancestrale inconnue.



Le modèle de Jukes et Cantor est réversible. Cela signifie que la dynamique d'une séquence est la même quand on regarde son évolution dans le passé ou dans le futur. Une telle propriété n'est bien sûr possible que si l'on travaille avec une séquence qui est déjà à l'équilibre, c'est à dire mélangée. En effet, imaginons qu'on laisse évoluer pendant un temps long une séquence ne comportant au départ que des adénines. La séquence obtenue est proche de l'état d'équilibre, en particulier la fréquence de chaque nucléotide est proche de $1/4$. Maintenant si on laisse évoluer cette séquence dans le passé avec la même dynamique, la fréquence de chaque nucléotide est toujours proche de $1/4$ et il est impossible de retomber sur notre séquence de départ.

Pour vérifier qu'une chaîne de Markov à temps continu est réversible, il suffit de vérifier le critère suivant.

Proposition 2.7 (Critère de Kolmogorov)**.** *Une chaîne de Markov à temps continu stationnaire est réversible si et seulement si les coefficients de son générateur Q vérifient*

$$q_{x_1x_2}q_{x_2x_3}\dots q_{x_{n-1}x_n}q_{x_nx_1} = q_{x_1x_n}q_{x_nx_{n-1}}\dots q_{x_3x_2}q_{x_2x_1},$$

pour tous éléments $x_1, x_2, \dots, x_n \in \mathscr{A}$.

Une conséquence de la réversibilité est la suivante : étant données deux séquences à l'équilibre, la probabilité d'observer les données est la même si l'on considère que ces deux séquences ont divergé depuis un temps t depuis une séquence ancestrale commune à l'équilibre ou bien que l'une a évolué à partir de l'autre pendant un temps $2t$.

Exercice XI (Modèle de Kimura, suite et fin)**.**

Pour estimer une distance phylogénétique dans le modèle de Kimura, on a encore une fois recours à la méthode du maximum de vraisemblance. En bonus, on peut aussi estimer le ratio des taux transition/transversion κ*. Les données sont résumées par le nombre de transitions* NS_{obs} *et le nombre de transversions* NV_{obs} *observables entre les deux séquences.*

$$S_{\text{obs}} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{1}\{\{X_i(0), X_i(t)\} = R\} + \mathbf{1}\{\{X_i(0), X_i(t)\} = Y\}\right),$$
$$V_{\text{obs}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\{X_i(t) \neq X_i(0)\} - S_{\text{obs}}.$$

Le nombre de sites inchangés est $N - NS_{\text{obs}} - NV_{\text{obs}}$*. Soient* D *et* K *définis par*

$$D = -\frac{1}{4}\log(1-2V_{\text{obs}}) - \frac{1}{2}\log(1-2S_{\text{obs}} - V_{\text{obs}}),$$
$$K = 2\frac{\log(1-2S_{\text{obs}} - V_{\text{obs}})}{\log(1-2V_{\text{obs}})} - 1.$$

Alors D *et* K *sont des estimateurs consistants de* $(w+2v)t$ *and* w/v*.*

1. *On prend comme séquence initiale une séquence de longueur* $N = 10000$ *ne comportant que des adénines. On fixe comme paramètres* $v = 0.25$ *et* $w = 0.5$*. Calculer* D *et* K *pour la séquence ayant évolué pendant les temps* $t = 0.1, 0.2, \ldots, 1$*.*

2.3 Modèles Markoviens de substitution de codons

Un codon est une séquence de trois nucléotides spécifiant l'un des 22 acides aminés protéinogènes dont la succession sur l'ARN messager détermine la structure primaire de la protéine à synthétiser. En tout, il y a donc 64 codons différents qui codent pour 22 acides aminés et l'arrêt de la synthèse (codon stop).

Nous présentons ici une version simplifiée du modèle de substitution de codons introduit par Goldman et Yang (1994). L'espace d'états de la chaîne est l'ensemble des codons $\mathscr{C}$ privés des codons stop puisqu'il ne sont pas utilisés à l'intérieur d'une protéine fonctionnelle. On a donc un espace d'états à 61 codons.

Le générateur $Q = \{q(x,y) : (x,y) \in \mathscr{C}^2\}$ est donné par
(2.4)

$$q(x,y) = \begin{cases} 0 & \text{si } x \text{ et } y \text{ diffèrent en deux ou trois sites,} \\ \pi_y & \text{si } x \text{ et } y \text{ diffèrent par une transversion synonyme,} \\ \kappa\pi_y & \text{si } x \text{ et } y \text{ diffèrent par une transition synonyme,} \\ \omega\pi_y & \text{si } x \text{ et } y \text{ diffèrent par une transversion non-synonyme,} \\ \omega\kappa\pi_y & \text{si } x \text{ et } y \text{ diffèrent par une transition non-synonyme,} \end{cases}$$

Substitution de `tcg` (Ser) par	Taux	Type de Substitution
`acg` (Thr)	$\omega\pi_{\texttt{acg}}$	Transversion non-synonyme
`ccg` (Pro)	$\omega\kappa\pi_{\texttt{ccg}}$	Transition non-synonyme
`gcg` (Ala)	$\omega\pi_{\texttt{gcg}}$	Transversion non-synonyme
`ttg` (Leu)	$\omega\kappa\pi_{\texttt{ttg}}$	Transition non-synonyme
`tgg` (Trp)	$\omega\pi_{\texttt{tgg}}$	Transversion non-synonyme
`tca` (Ser)	$\kappa\pi_{\texttt{tca}}$	Transition synonyme
`tct` (Ser)	$\pi_{\texttt{tct}}$	Transversion synonyme
`tcc` (Ser)	$\pi_{\texttt{tcc}}$	Transversion synonyme

TABLE 1 – Ensemble des voisins du codon `tcg` (Ser) dans le modèle présenté. On remarque que le codon `tag` provenant d'une substitution du nucléotide `c` en deuxième position par `a` n'est pas présent puisque c'est un codon stop.

où κ est le ratio transition/transversion, ω le ratio non-synonyme/synonyme et π_y la fréquence à l'équilibre du codon y. Les paramètres κ et π_y sont caractéristiques du processus d'évolution au niveau de l'ADN. En revanche, le paramètre ω sert à mesure la pression de sélection au niveau protéïque. En effet, si la sélection n'a pas d'impact, les mutations non-synonymes seront fixées autant que les mutations synonymes et $\omega = 1$. Si les mutations non-synonymes sont délétères, la sélection aura tendance à réduire leur taux de fixation et on aura $\omega < 1$. Si les mutations non-synonymes procurent un avantage sélectif, leur taux de fixation sera augmenté et on aura $\omega > 1$. Un taux ω significativement plus grand que 1 est caractéristique d'une évolution protéique procurant un avantage adaptatif.

Nous supposons toujours que les sites évoluent indépendamment des autres. Les coefficients de la matrice vérifient la condition de réversibilité

$$\pi_x q(x,y) = \pi_y q(y,x), \quad \forall x, y \in \mathscr{C}.$$

Ainsi, la log-vraisemblance associée au modèle et aux deux séquences observées est donnée par

$$\ell(t) = \sum_x \sum_y N(x,y) \log[\pi_x p_{xy}(t)], \tag{2.5}$$

avec $N(x,y)$ le nombre de sites comportant un codon x dans la séquence 1 et un codon y dans la séquence 2, et $p_{xy}(t)$ la probabilité de transition de x vers y au temps t donnée par $\mathrm{e}^{Qt}(x,y)$.

En appliquant la méthode du maximum de vraisemblance, on obtient ainsi des estimateurs $\hat{\kappa}$, $\hat{\omega}$ et $\hat{\pi}_y$ pour les paramètres κ, ω et π_y.

Définition 2.8. *Les nombres de substitutions synonymes et non-synonymes*

par codon, notés S_d et N_d, entre deux séquences sont définis par

$$S_d = t\rho_S = \sum_{x\neq y : aa_x = aa_y} \pi_x q(x,y)\, t$$
$$N_d = t\rho_N = \sum_{x\neq y : aa_x \neq aa_y} \pi_x q(x,y)\, t.$$

Les proportions de substitutions synonymes et non-synonymes sont notées ρ_S et ρ_N.

Définition 2.9. *Les nombres de sites synonymes et non-synonymes par codon, notés S et N, entre deux séquences sont définis par*

$$S = 3\rho_S^1 = \sum_{x\neq y : aa_x = aa_y} \pi_x q^1(x,y)\, t$$
$$N = 3\rho_N^1 = \sum_{x\neq y : aa_x \neq aa_y} \pi_x q^1(x,y)\, t,$$

avec $q^1(x,y)$ défini comme $q(x,y)$ avec $\omega = 1$ fixé.

Les proportions de mutations synonymes et non-synonymes sont notées ρ_S^1 et ρ_N^1.

Définition 2.10. *Le nombre de substitutions synonymes par site synonyme K_S ou d_S est donné par S_d/S. Le nombre de substitutions non-synonymes par site non-synonyme K_A ou d_N est donné par N_d/N.*

Ainsi, nous avons

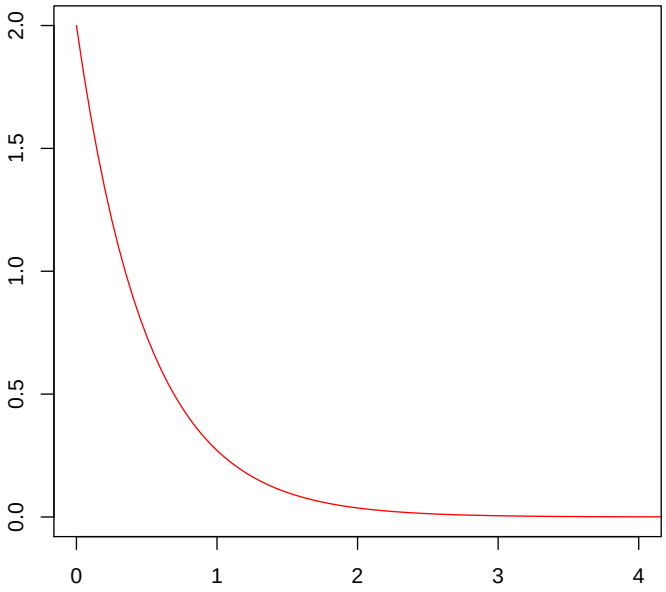
$$\omega = d_N/d_S = (\rho_N/\rho_S)/(\rho_N^1/\rho_S^1).$$

A Correction des exercices

Exercice I

1. D'après la définition 1.1, la densité d'une loi exponentielle de paramètre λ est donnée par $x \mapsto \lambda e^{-\lambda x}$. Pour tracer cette densité à l'aide de `R`, on peut soit rentrer directement la formule, soit utiliser la fonction `dexp`.

```
> lambda <- 2 ;
> x <- seq(0,10,length=200) ;
> plot(x,dexp(x,lambda),xlim=c(0,4),ylim=c(0,lambda),
+      col="red",type="lines",ann=FALSE )
```



2.

```
> mu <- 1 ;
> n <- 500 ;
> horloges <- rexp(n,mu) ;
```

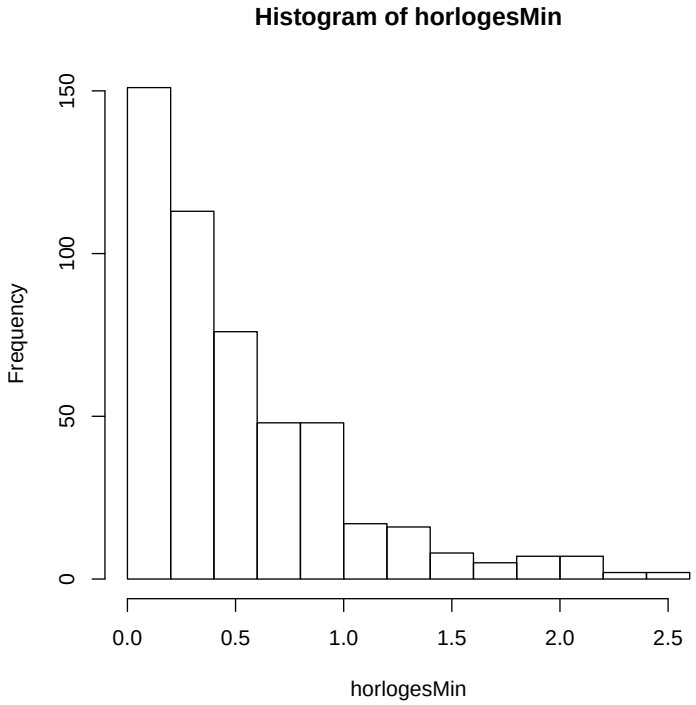
3.

```
> mu1 <- 1 ;
> mu2 <- 1 ;
> n <- 500 ;
> horloges1 <- rexp(n,mu1) ;
> horloges2 <- rexp(n,mu2) ;
> horlogesMin <- pmin(horloges1,horloges2) ;
```

4.

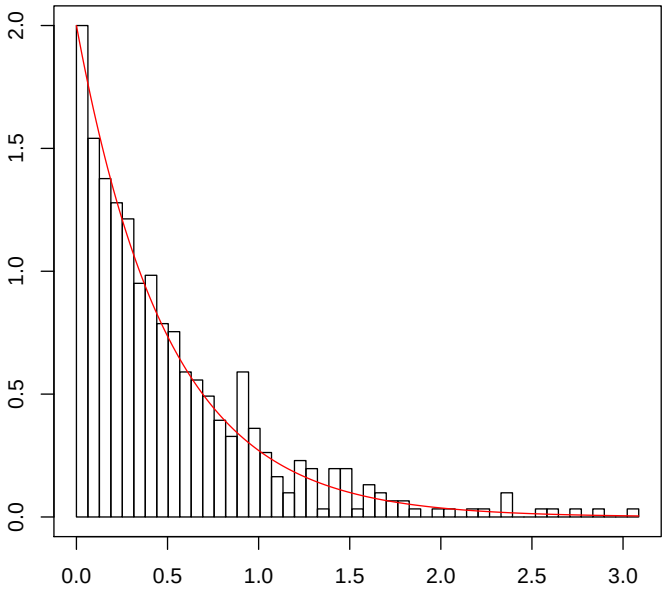
```
> mu1 <- 1 ;
> mu2 <- 1 ;
> n <- 500 ;
> horloges1 <- rexp(n,mu1) ;
```

```
> horloges2 <- rexp(n,mu2) ;
> horlogesMin <- pmin(horloges1,horloges2) ;
> hist(horlogesMin) ;
```

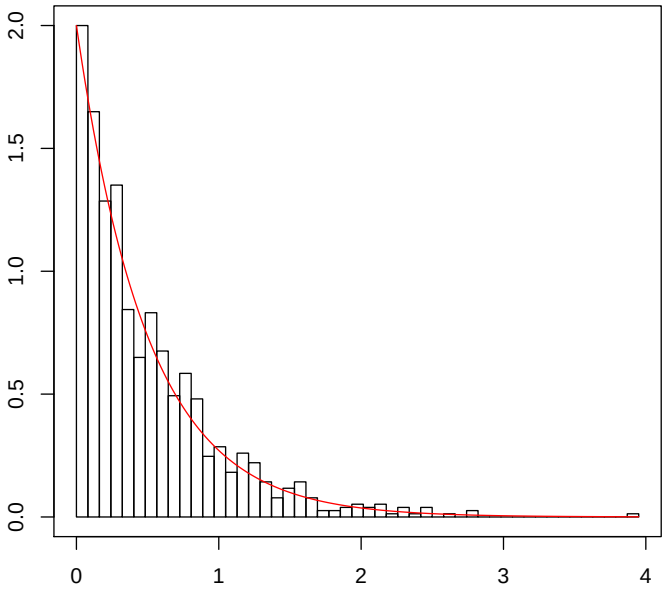


5.

```
> mu1 <- 1 ;
> mu2 <- 1 ;
> lambda <- 2 ;
> n <- 500 ;
> horloges1 <- rexp(n,mu1) ;
> horloges2 <- rexp(n,mu2) ;
> horlogesMin <- pmin(horloges1,horloges2) ;
> max <- max(horlogesMin)
> hist(horlogesMin,breaks = seq(0,max,length=50), xlim=c(0,max),
+ ann=FALSE,axes=FALSE) ;
> par(new=TRUE)
> x <- seq(0,max,length=200) ;
> plot(x,dexp(x,lambda),xlim=c(0,max),ylim=c(0,lambda),
+      col="red",type="lines",ann=FALSE )
```



6.

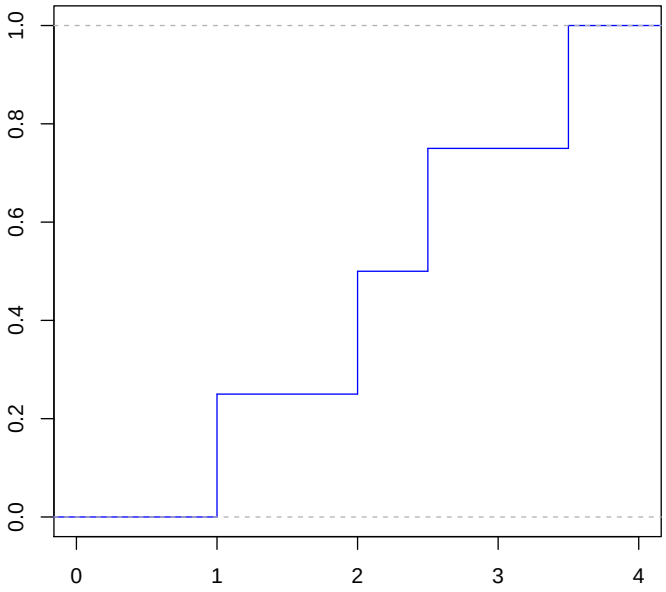


7. On constate que l'histogramme a tendance à épouser la courbe de la densité et que plus la taille de l'échantillon est importante, moins l'erreur est grande. Toutefois, on a le droit de rester dubitatif face à une telle figure.

Exercice II

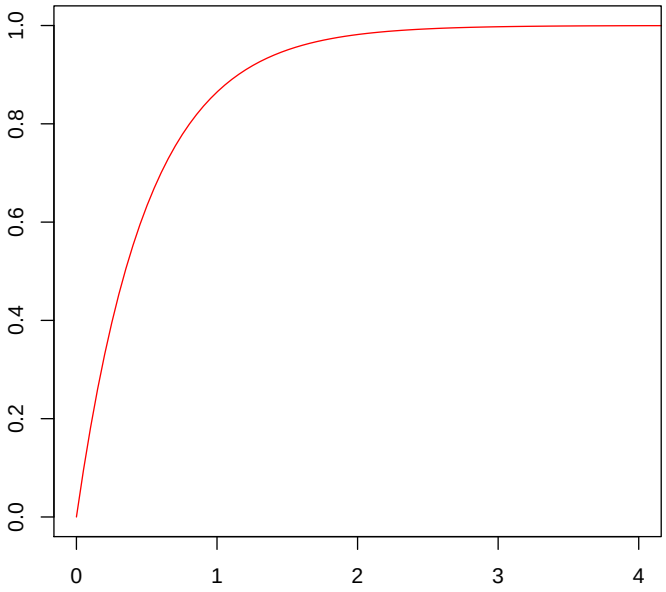
1.

```
> x <- c(2,1,2.5,3.5)
> Fn <- ecdf(x)
> plot(Fn, verticals=  TRUE, do.points = FALSE,
+      xlim=c(0,4),ylim=c(0,1),ann=FALSE,col="blue")
```



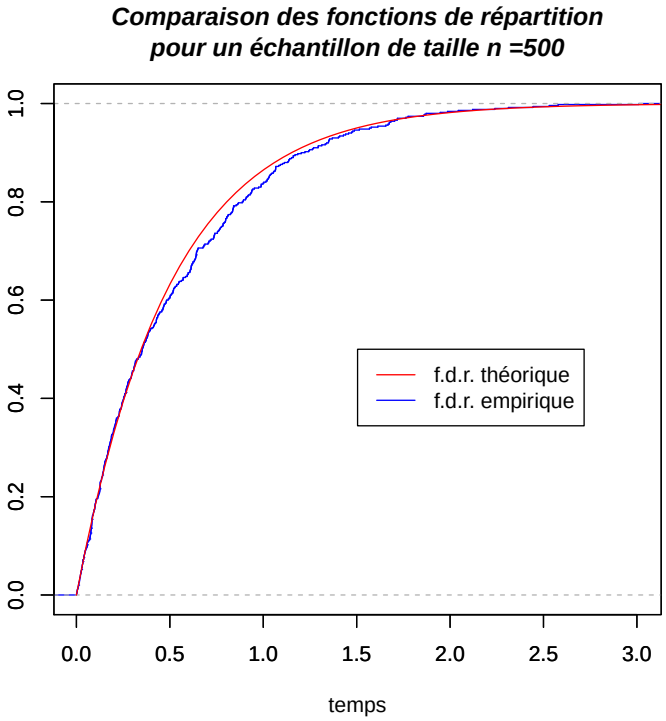
2. D'après la définition 1.1, la fonction de répartition d'une loi exponentielle de paramètre λ est donnée par $x \mapsto 1 - e^{-\lambda x}$. Pour tracer cette fonction de répartition à l'aide de `R`, on peut soit rentrer directement la formule, soit utiliser la fonction `pexp`.

```
> lambda <- 2 ;
> x <- seq(0,10,length=200) ;
> plot(x,pexp(x,lambda),xlim=c(0,4),ylim=c(0,1),
+      col="red",type="lines",ann=FALSE )
```



3.

```
> mu1 <- 1 ;
> mu2 <- 1 ;
> lambda <- 2 ;
> n <- 500 ;
> horloges1 <- rexp(n,mu1) ;
> horloges2 <- rexp(n,mu2) ;
> horlogesMin <- pmin(horloges1,horloges2) ;
> max <- max(horlogesMin)
> Fn <- ecdf(horlogesMin)
> plot(Fn, verticals=  TRUE, do.points = FALSE,
+      xlim=c(0,max),ylim=c(0,1),ann=FALSE,col="blue")
> par(new=TRUE)
> plot(x,pexp(x,lambda),xlim=c(0,max),ylim=c(0,1),
+      col="red",type="lines",ann=FALSE )
> legend(max/2, 0.5, c("f.d.r. théorique","f.d.r. empirique"),
+        cex=1, col=c("red","blue"),lty=c(1,1)) ;
> title(xlab="temps")
> title(main=paste("Comparaison des fonctions de répartition",
+         "\npour un échantillon de taille n =",n,sep=""),
+       col.main="black", font.main=4)
```



4.

```
> mu1 <- 1 ;
> mu2 <- 1 ;
> lambda <- 2 ;
> n <- 1000 ;
> horloges1 <- rexp(n,mu1) ;
> horloges2 <- rexp(n,mu2) ;
> horlogesMin <- pmin(horloges1,horloges2) ;
> max <- max(horlogesMin)
> Fn <- ecdf(horlogesMin)
> plot(Fn, verticals=  TRUE, do.points = FALSE,
+      xlim=c(0,max),ylim=c(0,1),ann=FALSE,col="blue")
> par(new=TRUE)
> plot(x,pexp(x,lambda),xlim=c(0,max),ylim=c(0,1),
+      col="red",type="lines",ann=FALSE )
> legend(max/2, 0.5, c("f.d.r. théorique","f.d.r. empirique"),
+        cex=1, col=c("red","blue"),lty=c(1,1)) ;
> title(xlab="temps")
> title(main=paste("Comparaison des fonctions de répartition",
+         "\npour un échantillon de taille n =",n,sep=""),
+       col.main="black", font.main=4)
```

5. Le théorème de Glivenko-Cantelli nous assure la convergence uniforme de la fonction de répartition empirique vers la fonction de répartition des variables aléatoires qui composent l'échantillon. Or on constate que quand la taille de l'échantillon est importante, la courbe bleue épouse la courbe rouge, on peut donc en conclure que la fonction de répartition des variables aléatoires de l'échantillon est celle d'une loi exponentielle de paramètre 2. Or l'échantillon a été simulé en prenant le minimum de deux lois exponentielles indépendantes de paramètre 1. On peut donc en conclure que le minimum de deux lois exponentielles de paramètre 1 suit une loi exponentielle de paramètre 2, ce qui est exactement l'objet de la propriété (iii) de la proposition 1.2.

Exercice III

1. D'après le théorème 1.5, le nombre de clients $N(t)$ arrivant dans un intervalle de longueur t suit une loi de Poisson de paramètre λt, d'où

$$\mathbb{P}(N(2) \geqslant 11) = 1 - \mathbb{P}(N(2) \leqslant 10) = 1 - \mathbb{P}(\mathrm{Poi}(10) \leqslant 10).$$

Pour calculer la valeur numérique de la quantité ci-dessus, on utilise le logiciel `R`. La fonction de répartition de la loi de Poisson de paramètre λ évaluée en x, c'est à dire $\mathbb{P}(\mathrm{Poi}(\lambda) \leqslant x)$, est donnée par `ppois(`$x,\lambda$`)`. Ici,

```
> 1 - ppois(10,10)

[1] 0.4169602
```

2. Avec nos notations, on cherche à calculer $\mathbb{P}(T_1 \leqslant t \,|\, N(0.5) = 1)$ pour $0 \leqslant t \leqslant 0.5$. Il vient

$$\begin{aligned}
\mathbb{P}(T_1 \leqslant t \,|\, N(0.5) = 1) &= \mathbb{P}(N(t) \geqslant 1 \,|\, N(0.5) = 1) \\
&= \frac{\mathbb{P}(N(t) = 1, N(0.5) - N(t) = 0)}{\mathbb{P}(N(0.5 = 1)} \\
&= \frac{\mathbb{P}(N(t) = 1)\mathbb{P}(N(0.5 - t) = 0)}{\mathbb{P}(N(0.5 = 1)} \\
&= \frac{\lambda t \mathrm{e}^{-\lambda t} \mathrm{e}^{\lambda(t-0.5)}}{0.5\lambda \mathrm{e}^{-0.5\lambda}} \\
&= 2t.
\end{aligned}$$

On voit donc que T_1 sachant $N(0.5) = 1$ suit la loi uniforme sur $[0, 0.5]$.

Exercice IV

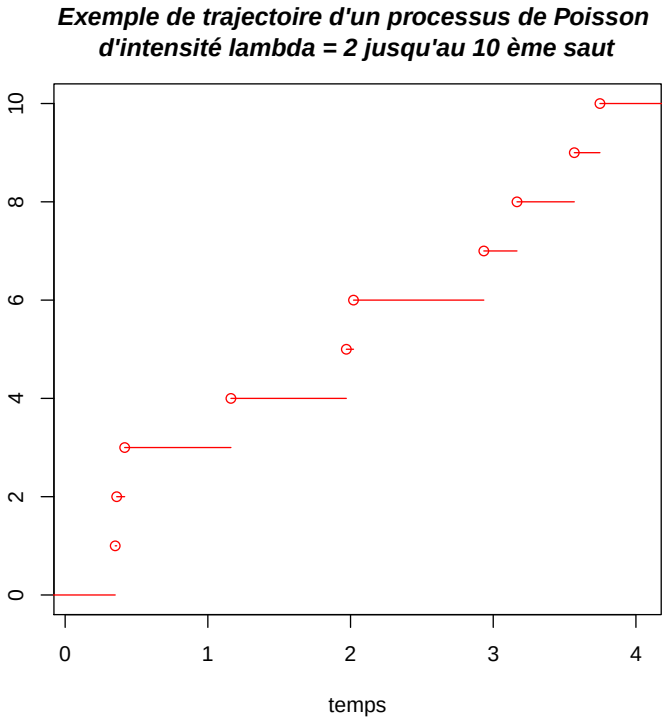
1. D'après le théorème 1.8, si les intervalles de temps entre deux évènements consécutifs d'un processus de comptage sont des variables aléatoires indépendantes obéissant à la même loi exponentielle de paramètre λ, alors le

processus est un processus de Poisson de paramètre λ. Pour simuler les n premiers temps d'arrivées d'un processus de Poisson d'intensité λ, il suffit donc de simuler n horloges suivant une loi exponentielle de paramètre λ, puis de constater que le premier temps d'arrivée correspond au temps écoulé sur la première horloge, que le deuxième temps d'arrivée correspond au temps écoulé sur la seconde horloge ajouté au temps écoulé sur la première horloge, etc.

```
> PoissonSaut <- function(n,lambda)
+ {
+   cumsum(rexp(n,lambda))
+ }
```

2.

```
> n <-10
> lambda <- 2
> t <- PoissonSaut(n,lambda) ;
> z <- seq(0,n,by=1)
> F <- stepfun(t,z)
> plot(F, verticals=  FALSE, ann=FALSE,  col="red")
> title(main=
+       paste("Exemple de trajectoire d'un processus de Poisson",
+         "\nd'intensité lambda =",lambda,"jusqu'au",
+             n,"ème saut",sep=" "),
+       col.main="black", font.main=4)
> title(xlab="temps")
```



3.

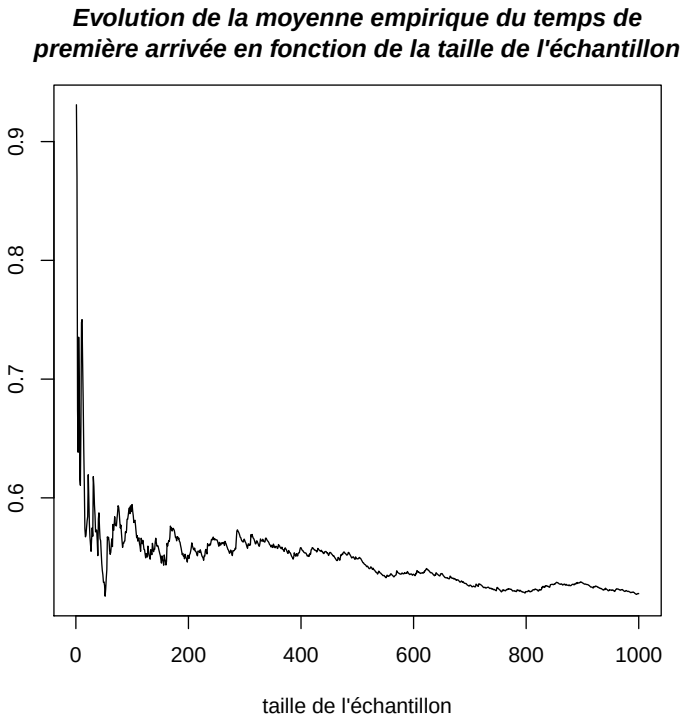
```
> n <-10 ;
> m <- 1000 ;
> lambda <- 2 ;
> matHorloges <- matrix(rexp(n*m,lambda),nrow=10,ncol=1000) ;
> matTemps <- apply(matHorloges,2,cumsum) ;
> ## pour verifier
> cumsum(matHorloges[,1]) - matTemps[,1] ;

 [1] 0 0 0 0 0 0 0 0 0 0
```

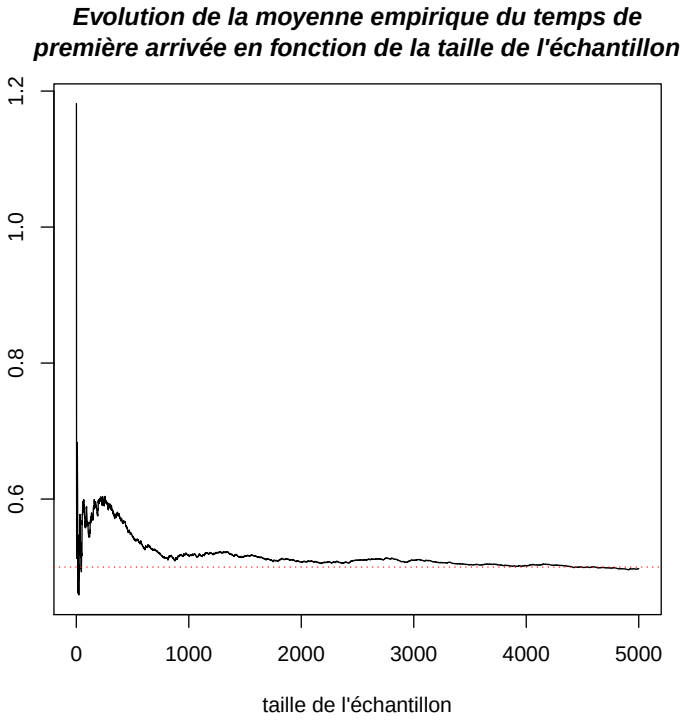
4.

```
> n <-10 ;
> m <- 1000 ;
> lambda <- 2 ;
> matHorloges <- matrix(rexp(n*m,lambda),nrow=n,ncol=m) ;
> matTemps <- apply(matHorloges,2,cumsum) ;
> moyenneTemps1 <- cumsum(matTemps[1,]) / (1 :m)  ;
> max <- max(moyenneTemps1)
```

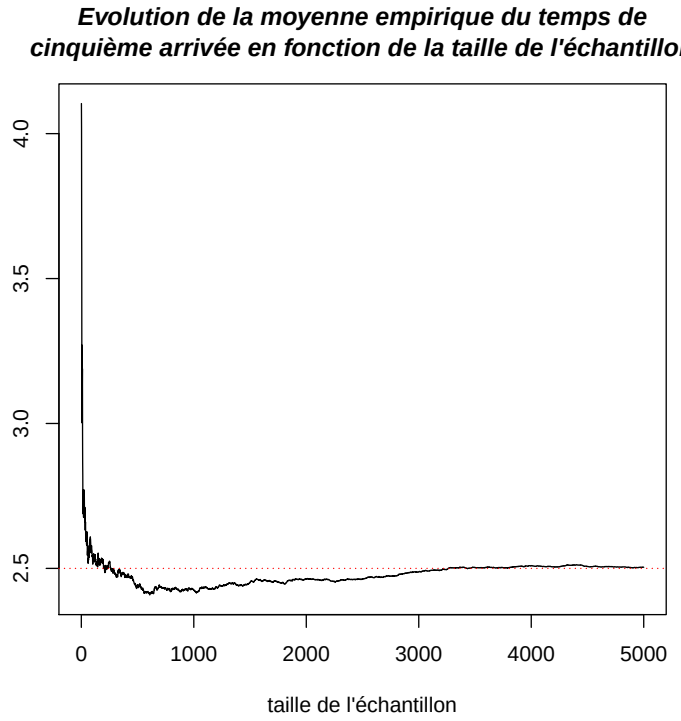
```
> min <- min(moyenneTemps1)
> plot((1 :m),moyenneTemps1,type="lines",ann=FALSE,ylim=c(min,max) )
> abline(h=1/lambda,col="red",lty=3)
> title(xlab="taille de l'échantillon")
> title(main=paste("Evolution de la moyenne empirique du temps de",
+         "\npremière arrivée en fonction de la taille de l'échantillon",sep=""),
+       col.main="black", font.main=4)
```



5.


Evolution de la moyenne empirique du temps de
première arrivée en fonction de la taille de l'échantillon
1.2
1.0
0.8
0.6
0
1000
2000
3000
4000
5000
taille de l'échantillon

6.



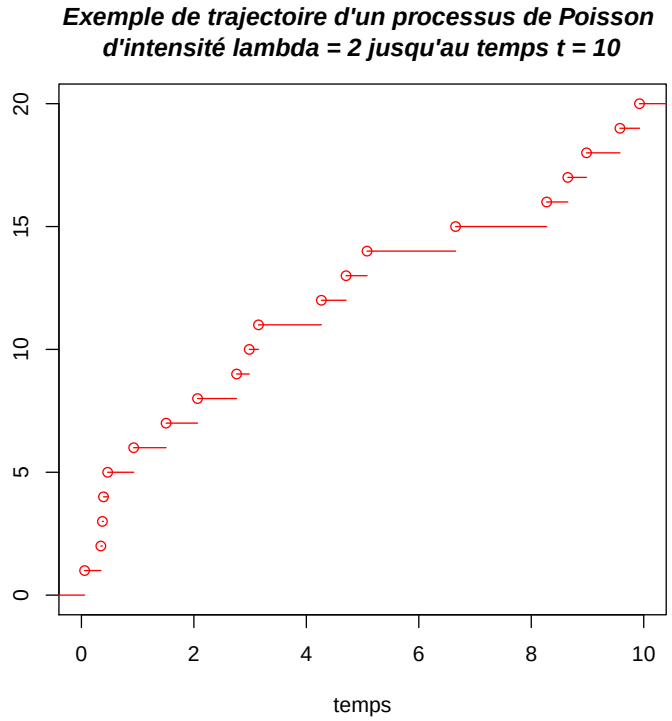
Exercice V

1.

```
> PoissonTemps <- function(t,lambda)
+ {
+ horloge  <- rexp(1,lambda)
+ newtemps <- horloge
+ temps <-newtemps
+ while( newtemps < t)
+ {
+ horloge <- rexp(1,lambda)
+ newtemps <- temps[length(temps)] + horloge
+ temps <- c(temps,newtemps)
+ }
+ temps[-length(temps)]
+ }
```

2. Simuler et afficher la trajectoire d'un processus de Poisson d'intensité $\lambda = 2$ jusqu'à l'instant $t = 10$.

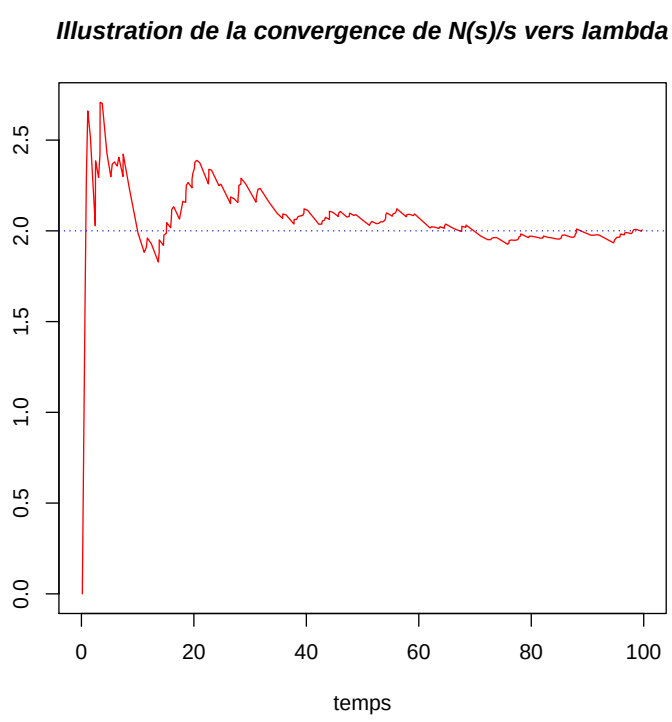
```
> t <- 10 ;
> lambda <- 2 ;
> x <- PoissonTemps(t,lambda)
> z<-seq(0,length(x),by=1)
> F <- stepfun(x,z)
> plot(F, verticals = FALSE, ann=FALSE, col="red",
+      xlim=c(0,t),ylim=c(0,max(z)))
> title(main=
+       paste("Exemple de trajectoire d'un processus de Poisson",
+         "\nd'intensité lambda =",lambda,"jusqu'au temps t =",
+             t,sep=" "), col.main="black", font.main=4)
> title(xlab="temps")
```



3.

```
> t <- 100 ;
> lambda <- 2 ;
> x <- PoissonTemps(t,lambda)
> z<-seq(0,length(x)-1,by=1)
> plot(x, z/x, ann=FALSE, col="red",
+      xlim=c(0,t),ylim=c(0,max(z/x)),type="lines")
> abline(h=lambda,col="blue",lty=3)
```

```
> title(main="Illustration de la convergence de N(s)/s vers lambda",
+       col.main="black", font.main=4)
> title(xlab="temps")
```



Exercice VI

1. D'après le théorème 1.5, la variable aléatoire $N(t)$ suit une loi de Poisson de paramètre λt.

```
> NombreSaut <- function(t,lambda)
+ {
+  rpois(1,lambda*t)
+ }
```

2.

```
> PoissonTemps2 <-  function(t,lambda)
+ {
+   n  <- NombreSaut(t,lambda)
+   if (n==0) t else sort( runif(n,0,t) )
+ }
```

3.

```
> t <-  10000
> lambda <-  2
> system.time(x1<-PoissonTemps(t,lambda),gcFirst = TRUE)

   user  system elapsed
  0.876   0.004   0.899

> system.time(x2<-PoissonTemps2(t,lambda),gcFirst = TRUE)

   user  system elapsed
  0.000   0.000   0.002
```

Exercice VII

1. D'après le cours, le paramètre de l'horloge exponentielle associée au nucléotide x est $-q(x,x)$. Dans le modèle de Jukes et Cantor, ce paramètre vaut 3λ indépendamment du nucléotide x. Ainsi, les temps inter-substitutions suivent tous une loi exponentielle de paramètre 3λ et le processus du nombre de susbtitutions au cours du temps est un processus de Poisson d'intensité 3λ.

2. D'après la question 1., on peut utiliser ce qui a été fait dans l'exercice VI pour la simulation.

```
> TempsSubstitutionJC <-  function(t,lambda)
+ {
+   n  <- rpois(1,3*lambda*t)
+   if (n==0) t else sort( runif(n,0,t) )
+ }
```

3.

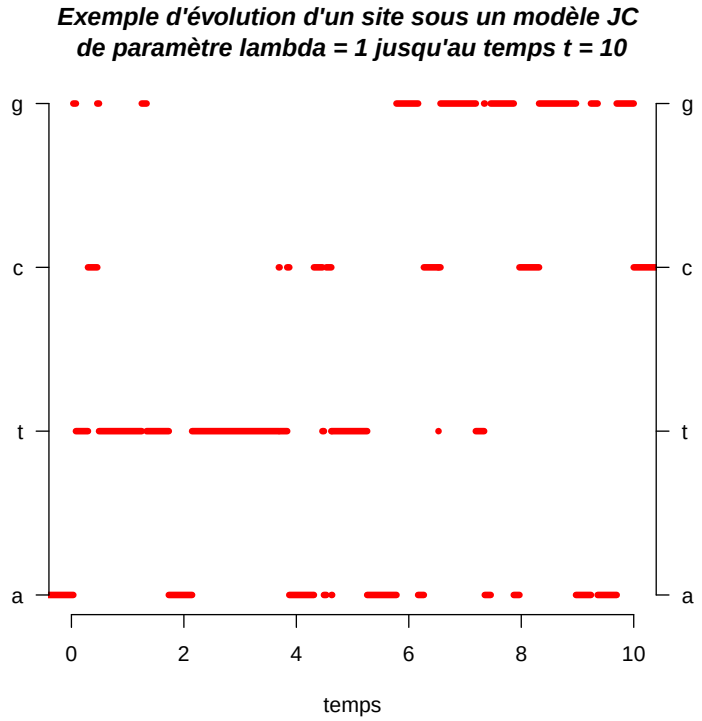
```
> numeronucleo <- 1 :4
> SubstitutionJC <-  function(i)
+ {
+   sample(numeronucleo[-i],1)
+ }
```

4.

```
> EvolutionJC <-  function(t,lambda,nuc)
+ {
+   numeronucleo <- 1 :4
+   suitenucleo <- nuc
+   x <- TempsSubstitutionJC(t,lambda)
+   n <- length(x)
+   if (n==0) matrix(c(t,1),nrow=2) else {
+   for (j in 1 :n)
+   {
+    suitenucleo <- c(suitenucleo, SubstitutionJC(suitenucleo[j]))
+    }
+   matrix(c(x,suitenucleo[-1]),nrow=2,byrow=TRUE)  }
+ }
```

5.

```
> t <- 10
> lambda <- 1
> nuc <- 1
> mat <- EvolutionJC(t,lambda,nuc)
> F <- stepfun(mat[1,],c(1,mat[2,]))
> plot(F, lwd=5,verticals = FALSE, do.points=FALSE,
+      ann=FALSE, col="red", xlim=c(0,t),ylim=c(1,4),axes=FALSE)
> title(main=
+       paste("Exemple d'évolution d'un site sous un modèle JC",
+         "\nde paramètre lambda =",lambda,"jusqu'au temps t =",
+             t,sep=" "), col.main="black", font.main=4)
> title(xlab="temps")
> axis(1, at=2*(0 :5))
> axis(2, at=1 :4, labels=c("a","t","c","g"),las=2)
> axis(4,at=1 :4, labels=c("a","t","c","g"),las=2)
```

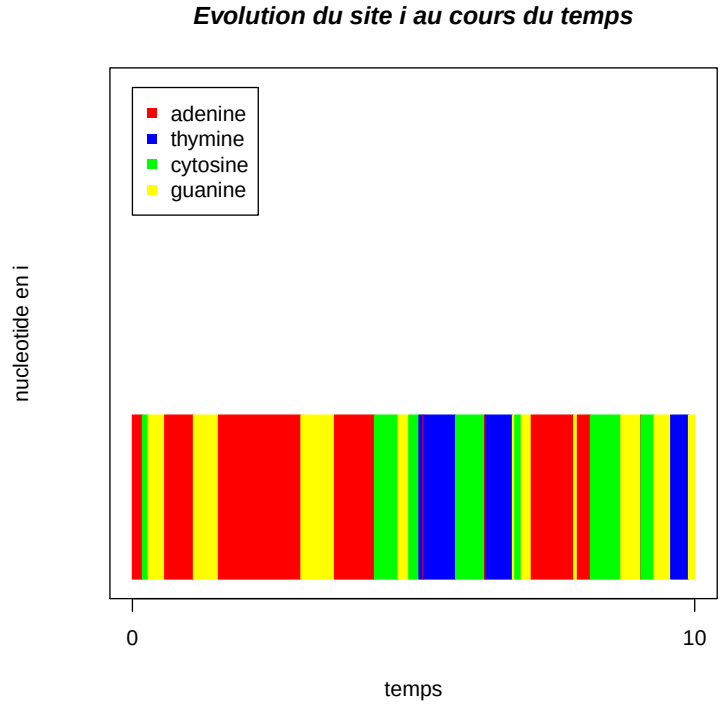


6.

```
> TempsSubstitutionJC <-  function(t,lambda)
+ {
+   n  <- rpois(1,3*lambda*t)
+   if (n==0) c(0,t) else c(0,sort( runif(n,0,t)),t )
+ }
> EvolutionJC <-  function(t,lambda,nuc)
+ {
+   x <- TempsSubstitutionJC(t,lambda)
+   suitenucleo <- nuc
+   n <- length(x)-2
+   if (n==0) matrix(c(x,1,1),nrow=2,byrow=TRUE) else {
+   for (j in 1 :n)
+   {
+    suitenucleo <- c(suitenucleo, SubstitutionJC(suitenucleo[j]))
+    }
+   matrix(c(x,suitenucleo,1),nrow=2,byrow=TRUE)  }
+ }
> t <- 10
> lambda <- 1
> nuc <- 1
```

```
> mat <- EvolutionJC(t,lambda,nuc)
> n <- length(mat[1,])
> z <- rep(1,n)
> couleurs <- c("red","blue","green","yellow")
> col <- couleurs[mat[2,][-n]]
> plot(mat[1,], z, type="n", xlim=c(0,t), ylim=c(0,3),
+      axes=FALSE, ann=FALSE,pch="|")
> rect(mat[1,][-(n)], z[-(n)] , mat[1,][-1], z[-1]-1,
+      col = col,border=col)
> axis(1,at=c(0,t), lab=c(0,t))
> box()
> legend(0, 3, c("adenine","thymine","cytosine","guanine"),
+        cex=1, col=c("red","blue","green","yellow"),
+        pch=c(15,15,15,15)) ;
> title(main="Evolution du site i au cours du temps",
+       col.main="black", font.main=4)
> title(xlab="temps")
> title(ylab="nucleotide en i")
> print(paste("Le nombre de substitutions ayant eu lieu est de ",
+ n-2,".", sep=""))

[1] "Le nombre de substitutions ayant eu lieu est de 29."
```



Exercice VIII

1.

```
> MarquageJC <- function(t,lambda,i)
+ {
+  n  <- rpois(1,lambda*t)
+   if (n==0) matrix(c(t,i),nrow=2) else {
+     matrix(c(sort( runif(n,0,t) ),rep(i,n)),nrow=2,byrow=TRUE)
+   }
+ }
```

2.

```
> EvolutionJC2 <- function(t,lambda)
+ {
+ mat <- MarquageJC(t,lambda,1)
+ for (i in 2 :4) { mat<- cbind(mat,MarquageJC(t,lambda,i)) }
+ mat[,order(mat[1,])]
+ }
```

3.

```
> t <-  10000
> lambda <-  1
> nuc <- 1
> system.time(mat1<-EvolutionJC(t,lambda,nuc),gcFirst = TRUE)

   user  system elapsed
  2.108   0.016   2.183

> system.time(mat2<-EvolutionJC2(t,lambda),gcFirst = TRUE)

   user  system elapsed
  0.012   0.000   0.011
```

Exercice IX

1. Comme la somme des coefficients par ligne d'un générateur est nulle, nous en déduisons que

$$Q = \begin{array}{c} \\ \mathtt{a} \\ \mathtt{t} \\ \mathtt{c} \\ \mathtt{g} \end{array} \begin{array}{c} \begin{array}{cccc} \mathtt{a} & \mathtt{t} & \mathtt{c} & \mathtt{g} \end{array} \\ \begin{pmatrix} -(2v+w) & v & v & w \\ v & -(2v+w) & w & v \\ v & w & -(2v+w) & v \\ w & v & v & -(2v+w) \end{pmatrix} \end{array}.$$

Le paramètre de l'horloge exponentielle associée à chaque nucléotide est donc $2v+w$. Ainsi, comme dans le modèle de Jukes et Cantor, les temps inter-substitutions ne dépendent pas de la nature du nucléotide qui occupe le site et suivent tous une loi exponentielle de paramètre $2v+w$.

2.

```
> TempsSubstitutionKimu <-  function(t,v,w)
+ {
+   n  <- rpois(1,(2*v+w)*t)
+   if (n==0) c() else sort( runif(n,0,t) )
+ }
```

3. Les probabilités que le nucléotide `a` soit remplacé par un `t`, un `c` ou un `g` sont données par $v/(2v+w)$, $v/(2v+w)$ et $w/(2v+w)$.

4. Écrivons le résultat sous la forme d'une matrice de transition.

$$
\begin{array}{c} \\ \mathtt{a} \\ \mathtt{t} \\ \mathtt{c} \\ \mathtt{g} \end{array}
\begin{array}{c}
\begin{array}{cccc} \mathtt{a} & \mathtt{t} & \mathtt{c} & \mathtt{g} \end{array} \\
\begin{pmatrix}
0 & v/(2v+w) & v/(2v+w) & w/(2v+w) \\
v/(2v+w) & 0 & w/(2v+w) & v/(2v+w) \\
v/(2v+w) & w/(2v+w) & 0 & v/(2v+w) \\
w/(2v+w) & v/(2v+w) & v/(2v+w) & 0
\end{pmatrix}
\end{array}. \tag{A.1}
$$

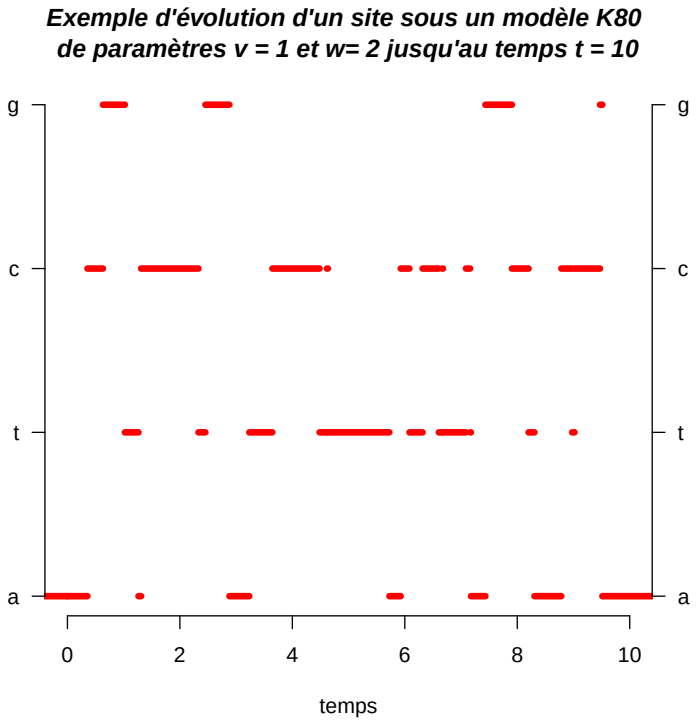
5.

```
> SubstitutionKimu <-  function(v,w,i)
+ {
+   numeronucleo <- 1 :4
+   P <- matrix(c(0,v,v,w,v,0,w,v,v,w,0,v,w,v,v,0)/(2*v+w),
+               nrow=4,ncol=4) ;
+   sample(1 :4,1,prob=P[i,])
+ }
```

6.

```
> EvolutionKimu <-  function(t,v,w,nuc)
+ {
+   tps <- c(0,TempsSubstitutionKimu(t,v,w))
+   suiteSubsti <- nuc
+   if (length(tps)  != 1 ){
+   for (j in 1 :(length(tps)-1))
+   {
+    suiteSubsti <- c(suiteSubsti, SubstitutionKimu(v,w,suiteSubsti[j]))
+    }
+   matrix(c(tps,suiteSubsti),nrow=2,byrow=TRUE)  }
+   else {
+   matrix(c(tps,suiteSubsti),nrow=2,byrow=TRUE)
+ }
+ }
> t <- 10
> v <- 1
> w <- 2
> nuc <- 1
> mat <- EvolutionKimu(t,v,w,1)
> F <- stepfun(mat[1,],c(1,mat[2,]))
> plot(F, lwd=5,verticals = FALSE, do.points=FALSE,ann=FALSE,
+      col="red", xlim=c(0,t),ylim=c(1,4),axes=FALSE)
> title(main=
+       paste("Exemple d'évolution d'un site sous un modèle K80",
```

```
+          "\nde paramètres v =",v,"et w=",w,"jusqu'au temps t =",
+              t, sep=" "), col.main="black", font.main=4)
> title(xlab="temps")
> axis(1, at=2*(0 :5))
> axis(2, at=1 :4, labels=c("a","t","c","g"),las=2)
> axis(4,at=1 :4, labels=c("a","t","c","g"),las=2)
```



Exercice X

1. Nous allons uniquement vérifier que le processus a les bonnes transitions $q(\mathtt{a},\mathtt{t})$ et $q(\mathtt{a},\mathtt{g})$ définis en (2.2) et les autres s'obtiennent de la même manière.

Soient $s>0$ petit et $t>0$ fixé. Comme les processus $\{M_x(t) : t \geqslant 0\}$ et $\{N_x(t) : t \geqslant 0\}$ sont indépendants, la probabilité d'avoir deux marques ou plus dans un intervalle de longueur s est de l'ordre de s^2. Les seules possibilités pour transformer un nucléotide en $\mathtt{t}$ sont de rencontrer une marque $M_{\mathtt{t}}$ ou une marque $N_{\mathtt{t}}$. Cependant, la marque $M_{\mathtt{t}}$ n'a aucun effet sur $\mathtt{a}$ puisque le remplacement de $\mathtt{a}$ par $\mathtt{t}$ n'est pas une transition. Ainsi, nous avons

$$\mathbb{P}(X_i(t+s)=\mathtt{t}|X_i(t)=\mathtt{a}) = \mathbb{P}(N_{\mathtt{t}}(t+s)-N_{\mathtt{t}}(t)=1)+s\varepsilon(s) = \nu s + s\tilde{\varepsilon}(s).$$

Les seules possibilités pour transformer un nucléotide en $\mathtt{g}$ sont de rencontrer une marque $M_{\mathtt{g}}$ ou une marque $N_{\mathtt{g}}$. Cette fois, la marque $M_{\mathtt{g}}$ a un effet sur $\mathtt{a}$

puisque le remplacement de a par g est une transition. Nous avons alors

$$\mathbb{P}(X_i(t+s) = \mathsf{g} | X_i(t) = \mathsf{a}) = \mathbb{P}(N_\mathsf{t}(t+s) - N_\mathsf{t}(t) = 1) + \mathbb{P}(M_\mathsf{t}(t+s) - M_\mathsf{t}(t) = 1) + s\varepsilon(s) = ws + \tilde{\varepsilon}(s).$$

2.

```
> MarquageV <- function(t,v,i)
+ {
+  n  <- rpois(1,v*t)
+   if (n==0){
+   data.frame(tps=t,nucleo=i,type='F')
+   }
+   else {
+   data.frame(tps=sort(runif(n,0,t)),nucleo=rep(i,n),
+              type=rep('V',n))
+   }
+ }
> MarquageW <- function(t,v,w,i)
+ {
+   n  <- rpois(1,(w-v)*t)
+   if (n==0){
+   data.frame(tps=t,nucleo=i,type='F')
+   }
+   else {
+   data.frame(tps=sort(runif(n,0,t)),nucleo=rep(i,n),
+              type=rep('W',n))
+ }
+ }
> EvolutionKimu2 <- function(t,v,w,nuc)
+ {
+ marques <- MarquageV(t,v,1)
+ for (i in 2 :4)
+   {
+     marques <- rbind(marques,MarquageV(t,v,i))
+   }
+ for (i in 1 :4)
+   {
+     marques <- rbind(marques,MarquageW(t,v,w,i))
+   }
+ marques <- marques[order(marques$tps) ,]
+ tempsSubsti <- 0
+ suiteSubsti <- nuc
+ for (i in 1 :nrow(marques)){
+   if (marques[i,]$type == 'V'){
```

```
+     tempsSubsti<-c(tempsSubsti,marques[i,]$tps)
+     suiteSubsti <- c(suiteSubsti,marques[i,]$nucleo)
+   }
+   if (marques[i,]$type == 'W'){
+     if ( (suiteSubsti[length(suiteSubsti)]
+           + marques[i,]$nucleo)==5 ){
+       tempsSubsti <- c(tempsSubsti,marques[i,]$tps)
+       suiteSubsti <- c(suiteSubsti,marques[i,]$nucleo)
+     }
+   }
+ }
+ matrix(c(tempsSubsti,suiteSubsti),nrow=2,byrow=TRUE)
+ }
```

3.

```
> t <-  100
> v <-  0.25
> w <- 0.5
> nuc <- 1
> system.time(mat1<-EvolutionKimu(t,v,w,nuc),gcFirst = TRUE)

   user  system elapsed
  0.004   0.000   0.002

> system.time(mat2<-EvolutionKimu2(t,v,w,nuc),gcFirst = TRUE)

   user  system elapsed
  0.100   0.000   0.118
```

4.

```
> EvolutionSite <- function(nuc,t,v,w)
+ {
+   mat <- EvolutionKimu(t,v,w,nuc)
+   mat[2,ncol(mat)]
+ }
```

5.

```
> EvolutionSequence <- function(seq,t,v,w)
+ {
+   sapply(seq,EvolutionSite,t=t,v=v,w=w)
```

```
+ }
> sequence <- c(1,2,2,3,4)
> EvolutionSequence(sequence,1,0.25,0.5)

[1] 4 2 2 3 4
```

6.

```
> N <- 10000
> pas <- 0.1
> v <- 0.25
> w <- 0.5
> sequence <- rep(1,N)
> mat <- c()
> for (k in 1 :40)
+   {
+     sequence <- EvolutionSequence(sequence,pas,v,w)
+     mat <- rbind(mat,c(mean(sequence==1),mean(sequence==2),
+                        mean(sequence==3),mean(sequence==4)))
+   }
> mat[c(1,10,20,30,40),]

       [,1]   [,2]   [,3]   [,4]
[1,] 0.9091 0.0239 0.0213 0.0457
[2,] 0.4585 0.1583 0.1558 0.2274
[3,] 0.3128 0.2213 0.2098 0.2561
[4,] 0.2774 0.2461 0.2290 0.2475
[5,] 0.2580 0.2483 0.2414 0.2523
```

Exercice XI

```
> N <- 10000
> pas <- 0.1
> v <- 0.25
> w <- 0.5
> sequence <- rep(1,N)
> d <- c()
> kappa <- c()
> for (k in 1 :10)
+   {
+     sequence <- EvolutionSequence(sequence,pas,v,w)
+     sObs <- mean(sequence == 4)
+     vObs <- mean(sequence  !=1 ) - sObs
```

```
+     d <- c(d,-1/4 * log(1-2*vObs) - 1/2 * log(1-2*sObs-vObs))
+     kappa <- c(kappa, 2*log(1-2*sObs-vObs)/log(1-2*vObs) - 1)
+   }
> data.frame(temps=0.1*(1 :10),D=d,kappa=kappa)

   temps         D    kappa
1    0.1 0.0989290 2.008632
2    0.2 0.2040389 1.970608
3    0.3 0.3048980 2.158204
4    0.4 0.4141007 2.063122
5    0.5 0.5210316 2.016759
6    0.6 0.6187345 2.016543
7    0.7 0.7112121 1.951717
8    0.8 0.8029197 1.953413
9    0.9 0.8969280 1.991038
10   1.0 0.9997525 2.022128
```

B Sujets d'examen

B.1 Première Session 2012–2013

M2 GBI, 1er semestre 2012-2013

Examen de Chaînes de Markov avancées

Jeudi 21 décembre 2012, de 9h00 à 12h00.

- Le sujet est une composition de mathématiques comportant des illustrations imposées à l'aide du logiciel `R`.
- Les questions ne comportant pas d'illustration sont à traiter sur une copie manuscrite. Une bonne rédaction est un élément d'appréciation important pour les copies.
- Tout au long du sujet, des noms de scripts et de fonctions sont imposés. Veillez à ne pas les modifier sous peine de ne pas être corrigé.
- Les illustrations seront regroupées dans un répertoire intitulé `ExamenCM2012_Nom_Prenom`, où vous indiquerez votre nom à la place de `Nom` et votre prénom à la place de `prenom`.

Début du sujet

On s'intéresse à la modélisation de la taille de la file d'attente de la poste de Plonéour-Lanvern dont le principe de fonctionnement est le suivant.

La poste ne comporte qu'un seul guichet qui ouvre à partir de 9h00. Dès cet horaire, une succession de clients se présente au guichet afin de bénéficier d'un service. Si le guichet est libre, la prise en charge du client au guichet commence immédiatement, sinon le client se place dans la file d'attente en attendant son tour.

Dans toute la suite, on note $\{X(t) : t \geqslant 0\}$ le processus qui compte le nombre de clients à l'intérieur de la poste. Ainsi, au temps $t \geqslant 0$, on comptabilise le client en train d'être servi (si il y en a un) et les clients dans la file d'attente (si il y en a une). Dans tout le problème, on fera les hypothèses suivantes.

(H1) L'arrivée des clients suit un processus de Poisson d'intensité $\lambda > 0$.

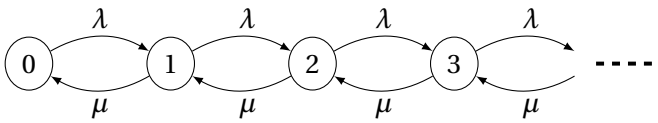
(H2) Les temps de service sont indépendants (et indépendants du processus d'arrivée des clients) et suivent la loi exponentielle de paramètre $\mu > 0$.

Partie I : compréhension du processus et simulation

On admet que le processus $\{X(t) : t \geqslant 0\}$ est une chaîne de Markov à temps continu, constant par morceaux (comme le processus de Poisson).

1. On suppose, dans cette question uniquement, que trois clients sont arrivés entre 9h00 et 9h35 aux horaires suivants : 9h08, 9h15 et 9h20. Leur temps de service ont été respectivement de 10, 4 et 7 minutes.
   (a) Représenter sur votre copie le graphe de l'évolution du nombre de clients à l'intérieur de la poste entre 9h00 et 9h35.
   (b) Écrire un script `I01b.R` qui donne le même graphe.
2. Quelles sont les valeurs possibles pour $X(t)$?
3. Lorsque le premier client arrive, quelle transition se produit pour le processus X ?
4. Rappeler la loi du temps d'arrivée du premier client. Sachant qu'aucun client n'est arrivé jusqu'au temps t, donner une probabilité approchée de l'arrivée du premier client dans l'intervalle de temps $]t, t+s]$ avec s petit.
5. On suppose qu'au temps $t > 0$ se trouvent n clients à l'intérieur de la poste avec $n \geqslant 1$, c'est à dire $X(t) = n$. Quels sont les évènements susceptibles de modifier X dans l'intervalle de temps $[t, t+s[$ avec s petit ? On donnera une probabilité approchée de ces évènements.

On admet dans la suite que le graphe de transition du processus X est le suivant.



6. Compléter le générateur infinitésimal Q du processus X ci-dessous.

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & \cdots \end{array} \\ \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{array}.$$

7. Donner le paramètre de l'horloge exponentielle associée à l'état 0. Qu'en est-il pour les autres états du processus ?

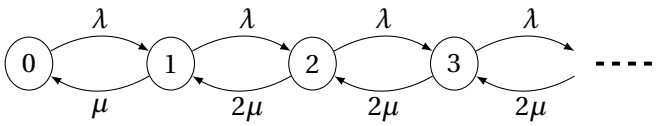
8. Écrire un script `I09.R` contenant une fonction `Horloge( entier, lambda, nu)` qui prend en entrée un entier `entier` et deux intensités `lambda` et `mu` et qui renvoie le paramètre de l'horloge exponentielle associée à l'état `entier`.
9. Soit $n \geqslant 0$. Donner les probabilités de transition de n vers les autres états en fonction de λ et μ (on distinguera en fonction de n).
10. Écrire un script `I10.R` contenant une fonction `Transition( entier, lambda, nu)` qui prend en entrée un entier `entier` et deux intensités `lambda` et `mu` et qui renvoie un entier choisi suivant les probabilités de transition établies en 9.
11. Écrire un script `I11.R` qui charge `I08.R` et `I10.R` et contenant une fonction `Trajectoire( entier, lambda, nu, temps)` qui prend en entrée un entier `entier`, deux intensités `lambda` et `mu` et un temps `temps`, et qui renvoie un tableau contenant les différents temps de sauts et états de la réalisation d'une trajectoire du processus X jusqu'au temps `temps`. On utilisera une boucle `while`.
12. Écrire un script `I12.R` qui charge `I11.R` et qui affiche une trajectoire du processus X avec les paramètres $\lambda = 10\text{h}^{-1}$ et $\mu = 6\text{h}^{-1}$ entre 9h00 et 12h00.

Partie II : faut il embaucher à Ploneour-Lanvern ?

Dans cette partie, on suppose que $\lambda > \mu$.

1. Intuitivement, que signifie l'hypothèse $\lambda > \mu$?
2. À l'aide de simulations dont le contenu sera placé dans un script appelé `II02.R`, déterminer la limite du rapport $\frac{X(t)}{t}$ en fonction de λ et μ.

On suppose dans la suite qu'un guichet supplémentaire est ouvert à la poste et on admet que cela modifie le graphe de transition du processus X de la manière suivante.



3. Écrire un script `II03.R` contenant de nouvelles fonctions `Horloge2`, `Transition2` et `Trajectoire2` afin de simuler la réalisation d'une trajectoire du processus X jusqu'à un temps `temps` avec un guichet supplémentaire.
4. Pour cette question, on suppose que $\lambda = 10\text{h}^{-1}$ et $\mu = 6\text{h}^{-1}$. En vous appuyant sur des simulations, étudier l'impact d'un guichet supplémentaire sur le processus.

Fin du sujet

M2 GBI, 1er semestre 2012-2013

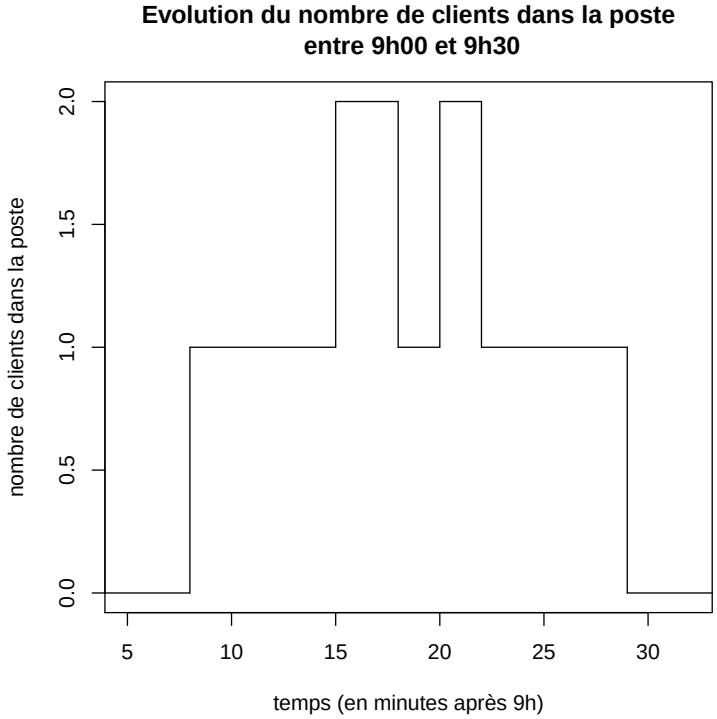
Corrigé Examen de Chaînes de Markov Avancées

Partie I : compréhension du processus et simulation

1.(a) Entre 9h00 et 9h08, il n'y a pas de clients dans la poste. A 9h08, le premier client arrive à la poste, sa prise en charge au guichet commence immédiatement et va durer dix minutes. A 9h15, le deuxième client arrive. Comme le premier client est au guichet jusqu'à 9h18, le deuxième client se place dans la file d'attente. Ainsi entre 9h08 et 9h15, un seul client se trouve à la poste, puis entre 9h15 et 9h18, il s'en trouve deux, un au guichet et un dans la file d'attente. À 9h18, le premier client quitte la poste et la prise en charge du deuxième client commence et va durer quatre minutes, c'est à dire jusqu'à 9h22. Comme le troisième client arrive à 9h20, on voit qu'il n'y a qu'un seul client à la poste entre 9h18 et 9h20, puis deux entre 9h20 et 9h22. À 9h22, la prise en charge du troisième client commence et va durer sept minutes. Comme aucun autre client n'arrive avant 9h30, on voit qu'il y a un client à la poste entre 9h22 et 9h29, puis aucun entre 9h29 et 9h30.

1.(b) Pour construire le graphe approprié, il faut représenter une fonction constante par morceaux. C'est la commande `stepfun` qui permet ceci comme on l'a vu dans l'exercice IV.

```
> temps<-c(8,15,18,20,22,29)
> nbclients<-c(0,1,2,1,2,1,0)
> plot(stepfun(temps,nbclients),
+      do.points=FALSE,
+      xlab="temps (en minutes après 9h)",
+      ylab="nombre de clients dans la poste",
+      main=paste("Evolution du nombre de clients dans la poste",
+        "\n entre 9h00 et 9h30",sep=""))
```



2. La quantité $X(t)$ représente le nombre de clients présents dans la poste au temps t et peut donc prendre n'importe quelle valeur entière positive ou nulle.

3. Lorsque le premier client arrive, le processus X passe de zéro à un.

4. Comme l'arrivée des clients suit un processus de Poisson d'intensité λ, on sait d'après le théorème 1.6 que le temps d'arrivée du premier client suit une loi exponentielle de paramètre λ. D'après la proposition 1.2, la probabilité de l'arrivée du premier client dans l'intervalle de temps $]t, t+s]$ avec s sachant qu'aucun client n'est arrivé jusqu'au temps t est donnée par $\lambda s[1+\varepsilon(s)]$ où $\varepsilon(s) \to 0$ quand $s \to 0$. Ainsi, quand s petit on peut approchée cette probabilité par λs.

5. Les évènements susceptibles de modifier le nombre de clients dans la poste dans l'intervalle de temps $[t, t+s]$ avec s petit sont l'arrivée d'un client ou la fin de la prise en charge du client au guichet. Le premier évènement se produit avec une proba d'environ λs d'après la propriété (H3) d'un processus de Poisson et le second avec une proba d'environ μs d'après le théorème 1.6.

6. Le générateur infinitésimal est donné par

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & \cdots \end{array} \\ \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \cdots \\ 0 & \mu & -(\lambda+\mu) & \lambda & \cdots \\ 0 & 0 & \mu & -(\lambda+\mu) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{array}.$$

7. D'après le cours pages 11 et 12, le paramètre de l'horloge exponentielle associée à l'état 0 est donné par $-q(0,0)$, c'est à dire λ. Le paramètre de l'horloge exponentielle associée à un état $x \neq 0$ est donné par $-q(x,x) = \lambda + \mu$.

8.

```
> Horloge=function(entier,lambda,mu){
+ if (entier == 0){
+ lambda
+ }
+ else {
+ lambda+mu
+ }
+ }
```

9. D'après le cours pages 11 et 12, la probabilité de transition d'un état x vers un état y est donné par $-q(x,y)/q(x,x)$. La probabilité de transition de 0 vers 1 est donc de 1 et la probabilité de 0 vers n est nulle pour tout $n \neq 1$. Si $n \geqslant 1$, alors la probabilité de transition de n vers $n-1$ est égale à $\mu/(\lambda+\mu)$, la probabilité de transition de n vers $n+1$ est égale à $\lambda/(\lambda+\mu)$ et les autres probabilités sont nulles.

10.

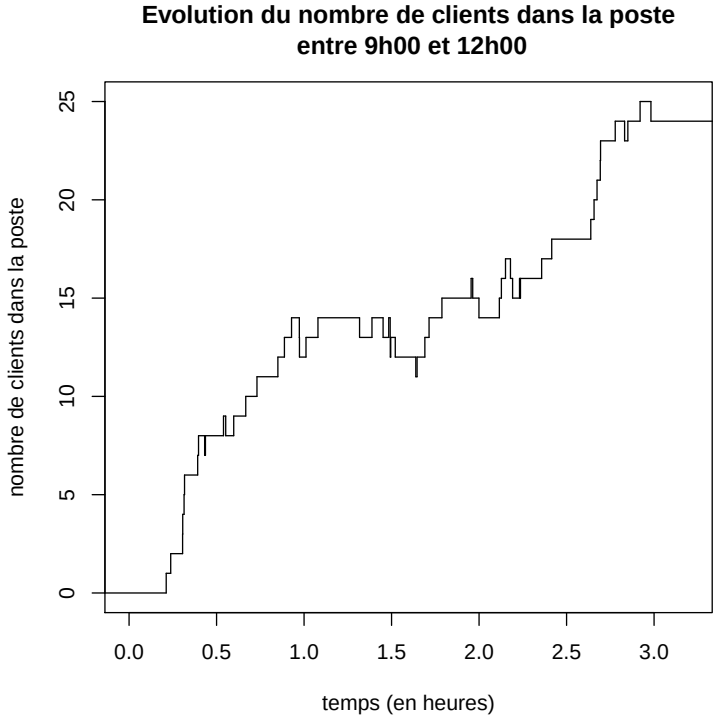
```
> Transition=function(entier,lambda,mu){
+ if (entier == 0){
+ 1
+ }
+ else {
+ sample(x=c(entier-1,entier+1),size=1,
+ prob=c(mu/(lambda+mu),lambda/(mu+lambda)))
+ }
+ }
```

11. En début de programme il faut ajouter `source("I09.R")` et `source("I10.R")` pour charger les scripts précédents.

```
> Trajectoire=function(entier,lambda,mu,temps){
+ tempssauts<-0
+ sauts<-entier
+ t<-0
+ while(t < temps){
+ parametre<-Horloge(entier,lambda,mu)
+ entier<-Transition(entier,lambda,mu)
+ t<-t+rexp(1,rate=parametre)
+ if (t<temps){
+         sauts<-c(sauts,entier)
+         tempssauts<-c(tempssauts,t)
+ }
+ }
+ data.frame(tempssauts,sauts)
+ }
```

12. En début de programme il faut ajouter `source("I11.R")` pour charger le programme précédent.

```
> trajet<-Trajectoire(0,10,6,3)
> plot(stepfun(trajet$tempssauts[-1],trajet$sauts),
+      do.points=FALSE,
+      xlab="temps (en heures)",ylab="nombre de clients dans la poste",
+      main=paste("Evolution du nombre de clients dans la poste",
+        "\n entre 9h00 et 12h00",sep=""))
```



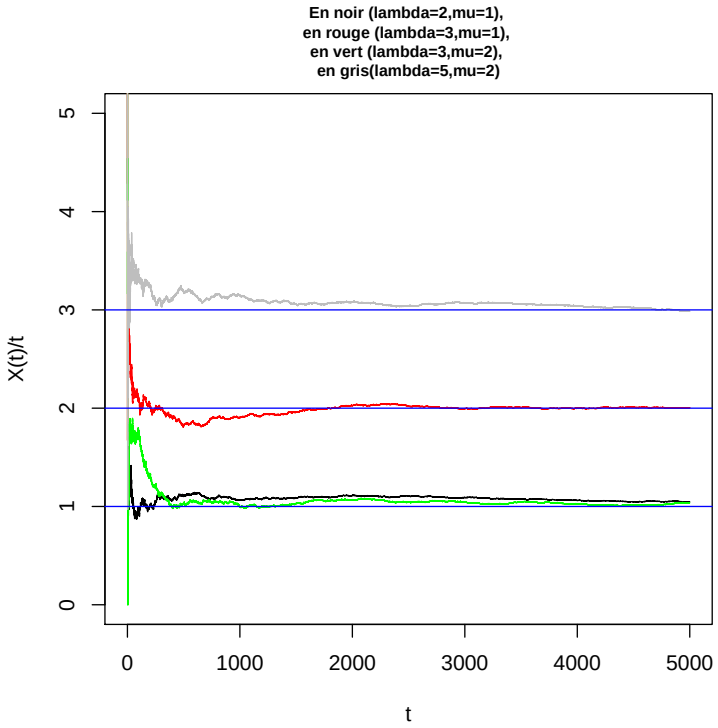
Partie II : faut il embaucher à Ploneour-Lanvern ?

1. Le fait que $\lambda > \mu$ signifie qu'en moyenne la prise en charge d'un client est plus longue que les temps inter-arrivées des clients.

2. À l'aide des simulations numériques suivantes, on peut penser que la limite de $X(t)/t$ vaut $\lambda - \mu$.

```
> tps <- 5000
> trajet1<-Trajectoire(0,2,1,tps)
> plot(y=trajet1$sauts/trajet1$tempssauts,
+      x=trajet1$tempssauts,type="l",
+      ylab="X(t)/t",xlab="t",ylim=c(0,5),ann=FALSE)
> title(main=paste("En noir (lambda=2,mu=1),",
+        "\nen rouge (lambda=3,mu=1),",
+        "\nen vert (lambda=3,mu=2),",
+        "\nen gris(lambda=5,mu=2)",sep=" "),cex.main=0.75)
> title(xlab="t")
> title(ylab="X(t)/t")
> trajet2<-Trajectoire(0,3,1,tps)
> lines(y=trajet2$sauts/trajet2$tempssauts,
+       x=trajet2$tempssauts,
+       ylim=c(0,5),type="l",col="red")
```

```
> trajet3<-Trajectoire(0,3,2,tps)
> lines(y=trajet3$sauts/trajet3$tempssauts,
+       x=trajet3$tempssauts,
+       ylim=c(0,5), type="l",col="green")
> trajet4<-Trajectoire(0,5,2,tps)
> lines(y=trajet4$sauts/trajet4$tempssauts,
+       x=trajet4$tempssauts,
+       ylim=c(0,5),type="l",col="grey")
> abline(h=1,col="blue")
> abline(h=2,col="blue")
> abline(h=3,col="blue")
```



3.

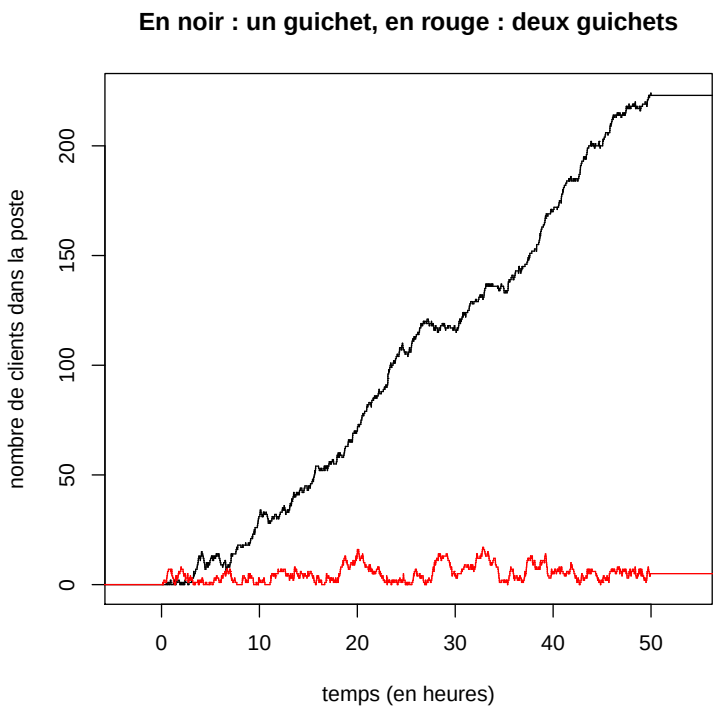
```
> Horloge2=function(entier,lambda,mu){
+ if (entier == 0){
+ lambda
+ }
+ else if (entier == 1){
+ lambda+mu
+ }
+ else {
```

```
+ lambda+2*mu
+ }
+ }
> Transition2=function(entier,lambda,mu){
+ if (entier == 0){
+ 1
+ }
+ else if (entier == 1){
+ sample(x=c(0,2),size=1,
+        prob=c(mu/(lambda+mu),lambda/(mu+lambda)))
+ }
+ else {
+ sample(x=c(entier-1,entier+1),size=1,
+        prob=c(2*mu/(lambda+2*mu),lambda/(2*mu+lambda)))
+ }
+ }
> Trajectoire2=function(entier,lambda,mu,temps){
+ tempssauts<-0
+ sauts<-entier
+ t<-0
+ while(t < temps){
+ parametre<-Horloge2(entier,lambda,mu)
+ entier<-Transition2(entier,lambda,mu)
+ t<-t+rexp(1,rate=parametre)
+ if (t<temps){
+          sauts<-c(sauts,entier)
+          tempssauts<-c(tempssauts,t)
+ }
+ }
+ data.frame(tempssauts,sauts)
+ }
>
> ### question 4)
```

4. Le guichet supplémentaire fait que le nombre de clients dans la poste ne croît plus linéairement en fonction du temps comme c'est le cas avec un seul guichet. En fait, avec un guichet supplémentaire, on diminue le temps d'attente moyen des clients dans la file d'attente et il est à présent inférieur au temps moyen inter-arrivée des clients. C'est très logique finalement.

```
> lambda<-10
> mu<-6
> trajet1<-Trajectoire(0,lambda,mu,50)
> trajet2<-Trajectoire2(0,lambda,mu,50)
```

```
> plot(stepfun(trajet1$tempssauts[-1],trajet1$sauts),
+      do.points=FALSE,
+      xlab="temps (en heures)",
+      ylab="nombre de clients dans la poste",
+      main="En noir  : un guichet, en rouge  : deux guichets")
> lines(stepfun(trajet2$tempssauts[-1],trajet2$sauts),
+       do.points=FALSE,col="red")
```



B.2 Seconde Session 2012–2013

M2 GBI, 1er semestre 2012-2013

Examen de Chaînes de Markov avancées, session 2

Mercredi 27 mars 2012, de 14h30 à 17h30.

- Le sujet est une composition de mathématiques comportant des illustrations imposées à l'aide du logiciel `R`.
- Les questions ne comportant pas d'illustration sont à traiter sur une copie manuscrite. Une bonne rédaction est un élément d'appréciation important pour les copies.
- Tout au long du sujet, des noms de scripts et de fonctions sont imposés. Veillez à ne pas les modifier sous peine de ne pas être corrigé.
- Les illustrations seront regroupées dans un répertoire intitulé `ExamCM2012_S2_Nom_Prenom`, où vous indiquerez votre nom à la place de `Nom` et votre prénom à la place de `Prenom`.

Début du sujet

Le long d'une route, l'écoulement (unidirectionnel) des véhicules peut être décrit par un processus de Poisson de paramètre $\lambda = 2$, noté $\{N(t) : t \geqslant 0\}$. Pour rappel, $N(t)$ représente le nombre de véhicules qui sont passés jusqu'au temps t.

On admet que chaque véhicule est soumis à une expérience de Bernoulli qui l'attribue avec probabilité $p = 60\%$ au type 1 et $q = 1 - p = 40\%$ au type 2, où
- les véhicules de type 1 mesurent 5 mètres de longueur (voitures légères) ;
- les véhicules de type 2 mesurent 10 mètres de longueur (cars, camions).

On note $\{N_i(t) : t \geqslant 0\}$ le processus formé des évènements du type i $(i = 1, 2)$. Ainsi, $N_i(t)$ représente le nombre de véhicules de type i qui sont passés jusqu'au temps t.

Partie I : compréhension du processus et simulation

1. Que peut-on dire des processus N_1 et N_2 ?
2. Soit $t \geqslant 0$. Comment s'écrit $N(t)$ en fonction de $N_1(t)$ et $N_2(t)$?
3. (a) Écrire un script `I03a.R` contenant une fonction `NombreVehicule (temps)` qui prend en entrée un temps `temps` et qui renvoie le nombre de véhicules qui sont passés sur la route jusqu'au temps `temps`.
   (b) Écrire un script `I03b.R` qui charge `I03a.R` et contenant une fonction `TempsVehicule(temps)` qui renvoie un tableau contenant les temps de passage des véhicules qui sont passés sur la route jusqu'au temps `temps`, sans utiliser de boucle `while`.
   (c) Écrire un script `I03c.R` qui charge `I03b.R` et qui affiche la trajectoire du processus de passage des véhicules jusqu'au temps $t = 10$.
4. (a) Écrire un script `I04a.R` contenant une fonction `NombreVehicule-Type (temps,type)` qui prend en entrée un temps `temps` et un type `type` et qui renvoie le nombre de véhicules de type `type` qui sont passés sur la route jusqu'au temps `temps`.
   (b) Écrire un script `I04b.R` qui charge `I04a.R` et contenant une fonction `TempsVehiculeType(temps,type)` qui renvoie un tableau contenant les temps de passage des véhicules de type `type` qui sont passés sur la route jusqu'au temps `temps`, sans utiliser de boucle `while`.
   (c) Écrire un script `I04c.R` qui charge `I04b.R` et qui affiche sur le même graphique les trajectoires des processus de passage des véhicules de type 1 et 2 jusqu'au temps $t = 10$.

Partie II : interruption du trafic

À cause de travaux, le trafic est interrompu alternativement pour chaque direction pendant un temps t_0 fixé. On note $\{L(t) : 0 \leqslant t \leqslant t_0\}$ le processus de la longueur de la queue formée par les véhicules au cours d'une interruption de durée t_0.

1. On suppose pour cette question uniquement que $t_0 = 10$ et que trois véhicules de type 1 sont arrivés aux temps 1, 4 et 8, et deux véhicules de type 2 aux temps 2 et 5.
   (a) Représenter sur votre copie le graphe de l'évolution de la longueur de la queue au cours du temps jusqu'à $t_0 = 10$.
   (b) Écrire un script `II01b.R` qui donne le même graphe.
2. Donner la longueur de la queue $L(t)$ au temps t en fonction de $N_1(t)$ et $N_2(t)$.

3. (a) Écrire un script `II03a.R` s'inspirant de`I04b.R` et contenant une fonction `Arrivee(temps)` qui prend en entrée un temps `temps` et qui renvoie un tableau contenant les temps d'arrivées à la zone des travaux des véhicules ainsi que leur type jusqu'au temps `temps`.

   (b) Écrire un script `II03b.R` qui charge `II03a.R` et contenant une fonction `Longueur(temps)` qui prend en entrée un temps `temps` et qui renvoie la trajectoire de l'évolution de la longueur de la queue jusqu'au temps `temps`.

4. Quel est le nombre maximal de véhicules de type 1 que la queue peut contenir sans que sa longueur ne dépasse 50 mètres ?

5. Quel est le nombre maximal de véhicules de type 2 que la queue peut contenir sans que sa longueur ne dépasse 50 mètres ?

6. Donner l'ensemble des valeurs possibles pour $N_1(t_0)$ et $N_2(t_0)$ pour que la longueur de la queue ne dépasse pas 50 mètres après une interruption d'une durée t_0.

7. (a) Écrire un script `II07a.R` contenant une fonction `File50(temps)` qui prend en entrée un temps `temps` et qui renvoie la probabilité que la longueur de la file au temps `temps` soit inférieure à 50 mètres.

   (b) Écrire un script `II07b.R` qui charge `II07a.R` et qui affiche les probabilités que la longueur de la file aux temps $t = 0, 0.5, 1, \ldots, 10$ soit inférieure à 50 mètres.

Fin du sujet