

Algorithmique et programmation

Rapport de mini-projet n°2

Victor GOURAIN - Clément Niel

Abdelhalim Larhlimi

Master 1 Bio-informatique et bio-statistique

2012-2013

- **Sommaire**

I. Le programme principal.....	2
II. Les fonctions.....	2
1.Fonctions de gestion de la matrice P.....	2
2.Fonctions de gestion de l'historique.....	3
III. Perspectives.....	4

I. Le programme principal

Le programme principal est développé de manière à modéliser au mieux la réalité. Il se base donc sur une expérience de culture cellulaire réalisée en laboratoire.

Nous posons une itération comme étant le temps nécessaire à une cellule pour se diviser soit l'équivalent de 24h. En laboratoire, les cultures cellulaires ne dépassent que très rarement deux mois (60 itérations) et sont réinitialisée au bout de 10 jours. Par réinitialisé, il faut entendre que 100 000 cellules sont prélevées de la culture courante pour former une nouvelle culture. Dans ce cas, l'âge des cellules est considéré comme étant réinitialisé.

La matrice P est de taille carrée de côté fixée à 50 cases. Cela allège l'affichage de la matrice. Pour mieux approcher la réalité les variables α , β , γ et δ sont définissables par l'utilisateur mais ont également des valeurs par défaut. Par défaut, le nombre d'itérations $p=1000$, α est égal à 3 (soit 3 jours, période d'acclimatation des cellules). β est défini à 0,5, γ est fixé à 100 et δ à 0,8.

Pour développer un programme proche de la réalité biologique, lors de la division d'une cellule, la cellule mère ne change pas de place et la cellule fille, qui est plus jeune que la cellule mère, est positionnée de manière aléatoire dans une case vide aux alentours.

Pour faciliter le comptage et la navigation dans la matrice P, les bords de la matrice P sont constitués de pointeurs de type cellule vide. Leur âge ne varie pas et elles ne peuvent ni se diviser, ni se transformer, ni mourir. Elles restent inchangées durant tout le déroulement du programme principal.

Un fichier Makefile a été créé pour faciliter la compilation.

II. Les fonctions

1. Fonctions de gestion de la matrice P

- Initialisation de la matrice P

La fonction initMat permet d'initialiser la matrice de pointeurs P : chaque pointeur est initialisé à NULL.

La fonction initPosition répartit de manière aléatoire les cellules. Elle appelle la fonction alloueCell. Cette dernière initialise une cellule à $type=0$, $age=0$, et $nombre\ de\ voisins=0$.

La fonction afficheMat est une fonction supplémentaire qui permet d'afficher la matrice à l'état courant. Pour chaque case de la matrice, elle affiche « o » s'il y a une cellule saine et « * » si la cellule est cancéreuse.

La fonction bordures permet de définir les bordures de la matrice P. L'utilisateur n'aura aucune connaissance de ces bordures. Cette fonction est utile pour le comptage des voisines et la division des cellules.

- Evolution de la matrice P

La fonction tuer_f_Cancer permet de supprimer (tuer) une cellule si parmi ses voisines il y a une cellule cancéreuse. Cette fonction prend en paramètre le facteur de risque δ (delta).

La fonction tuer_f_nbVoisines permet de supprimer une cellule si son nombre de voisines est strictement supérieur à 4.

La fonction incrementeAge fait vieillir les cellules à chaque itération et la fonction chgmtType, en fonction de la probabilité β (beta) et de l'âge de la cellule, la transforme en cellule cancéreuse.

La fonction divisions permet aux cellules ayant atteint l'âge de maturité de se diviser. La cellule mère ne change pas de position dans la matrice et la cellule fille est positionnée de manière aléatoire parmi les cases vides autour de la cellule mère.

La fonction nbHasard permet de définir un nombre au hasard qui est -1, 0 ou 1. Cela permet de définir aléatoirement une position autour d'une cellule, c'est pourquoi cette fonction est appelée, par exemple, lors du positionnement de la cellule fille après une division.

La fonction comptVoisines permet de compter les voisines de toutes les cellules de la matrice P.

La fonction compttype permet de compter les cellules de la matrice P à l'état courant, selon leur type.

2. Fonctions de gestion de l'historique

- Initialisation et incrémentation de la matrice H

Allouematrice est une fonction qui permet de créer et d'allouer l'espace mémoire de la matrice `matH` de type `H`. `matH` comporte un tableau bidimensionnel de pointeurs pointant vers un poiteur de type de type `HCASE`.

InsererHCase est une fonction qui permet d'insérer un objet de liste chaînée d'historique de case. Elle est appelée à chaque évènement.

AlloueHCase est appelée pour chaque allocation de l'espace mémoire d'un `hcase`. DefHCase permet de définir un `hcase` (n° d'itération, type d'évènement, poiteur vers objet suivant).

- Sauvegarde sur fichier

A la fin de la dernière itération la fonction sauvegarde est appelée par la programme principal pour sauvegarder dans le fichier `historique.txt` le contenu de la matrice `H`. Le fichier s'organise de la façon suivante :

1-1

|1| Naissance

|34| Transformation

Cette exemple illustre pour la case 1 de la ligne 1 une naissance à l'itération n°1 suivie par d'une transformation à l'itération n°34.

III. Perspectives

Le programme sera modifié pour permettre à l'utilisateur de définir la taille de son milieu de culture.