Traitement d'Images Numériques :une introduction avec le logiciel ImageJ et le langage Java

Tout le matériel nécessaire se trouve soit sur http://www.math-info.univ-paris5.fr/~lomn/Cours/CV/TI/Material/Tutorial/ ou ../Material/Data/. Pour ce TP, on aura besoin d'un JRE (pour l'exécution de code Java et l'exécutable *java*) et d'un JDK (pour la compilation java et l'exécutable *javac*) installés sur vos machines.

Prendre contact avec le langage Java

Dans le cours, on utilise la classe (concept de la programmation orienté objet) ***Threshold.class*** . Cette classe est obtenue en compilant le code java ***Threshold.java*** qui utilise deux autres classes ***Luminance.java*** et ***Picture.java***.
Ces trois classes sont une bonne base pour s'amuser un peu avec des images et étudier de façon basique la structure d'un programme écrit en langage Java.
Ces trois classes sont très bien commentées mais je reprends ici les étapes pour les utiliser.
Le grand avantage de Java est sa portabilité : un pseudo-code compilé de type *.class* est interprétable par tous les systèmes d'exploitation car il se base sur un *Java Runtime Environment* qui est installé par défaut normalement sur les machines actuelles.

Ai-je un *jre* et quelle version ?
Pour vérifier la présence de *jre* et sa localisation sur votre système utiliser les commandes Unix : soit ***$locate *jre**** soit ***$find /usr -name *jre**** par exemple (et à adapter comme toujours).
(Si pas présent, installez un en environnement de développement Java JDK (Java Development Kit : apt-get ***install*** *default-****jdk****)*
Récupérez d'abord seulement les trois fichiers ***.class*** pré-compilés***: Threshold.class, Luminance.class et Picture.class.***

$java Picture mandrill.jpg

Cette commande actionne le JRE et cherche la fonction main() dans la classe Picture afin d'afficher l'image ***mandrill.jpg***.

On va utiliser la classe principale ***Threshold*** qui fait un premier traitement de base avant de pouvoir par exemple compter automatiquement le nombre de cellules dans une image de façon automatique.

$java Threshold mandrill.jpg

On va travailler sur l'image ***embryos.jpg.*** Seuillez cette image avec la classe précédente. Obtient-on une segmentation correcte fond/objets d'intérêt ?

Récupérez à présent les codes source ***.java*** de ces trois classes: ***Threshold.java, Luminance.java et Picture.java.*** On va pouvoir créer nos propres programmes:-)

Que fait la classe Luminance ? (Ouvrez un éditeur de texte ou de code et lisez le code ***Luminance.java)***.

Et quel seuil utilise le code ***Threshold.java*** ?
Essayez un seuil plus élevé ? Un seuil moins élevé ? Pour cela on va recompiler le code java pour obtenir une nouvelle classe ***.class*** exécutable par le JRE de votre machine. Mais avant cela repérer l'exécutable de compilation ***javac*** dans votre arborescence (commande unix ***$locate javac***).
*(Si pas présent, installez un en environnement de développement Java JDK (Java Development Kit : apt-get **install** default-**jdk)***

1. Modifier la valeur du seuil THRESHOLD
2. Compiler le nouveau code java (sauvegardé)
   $javac Threshold.java

Remarque : Pour les futurs développeurs, un environnement tel que Java Development Kit (JDK) est conseillé (http://www.oracle.com/technetwork/java/javase/downloads/index.html).
Cet environnement fournira notamment la librairie permettant (facilement) de compiler des plugins imageJ en Java pour les modifier**.**

3. Testez en exécutant votre classe ***Threshold.class*** (vérifiez sa nouveauté avec la commande unix ***$ls -als***).
   $java Threshold embryos.jpg

Remarques : comme la classe ***Threshold.class*** dépend de ***Picture*** et ***Luminance*** il faut que ces deux classes soient présentes dans votre répertoire de travail.

Bonus : play with Fibonnacci.java files and uses top unix command to observe CPU. (why not valgrind and massif visualizer as well)

Lien avec le langage Python

Dans le répertoire Python, vous avez une trame de départ pour réaliser une fonction de seuillage ***Threshold*** avec un paramètre ou deux au choix. A vous de jouer.

Réflexion (pour plus tard) :

Créer la fonction composantes connexes ***Connexe*** qui permet de labelliser comme illustrée ci dessous les objets une fois l'image seuillée. Imaginez simplement de scanner l'image de haut en bas et de gauche à droite puis de propager un label aux voisins si le pixel est blanc et non déjà labellisé. https://www.lri.fr/~cabaret/ECC-in-a-nutshell/