

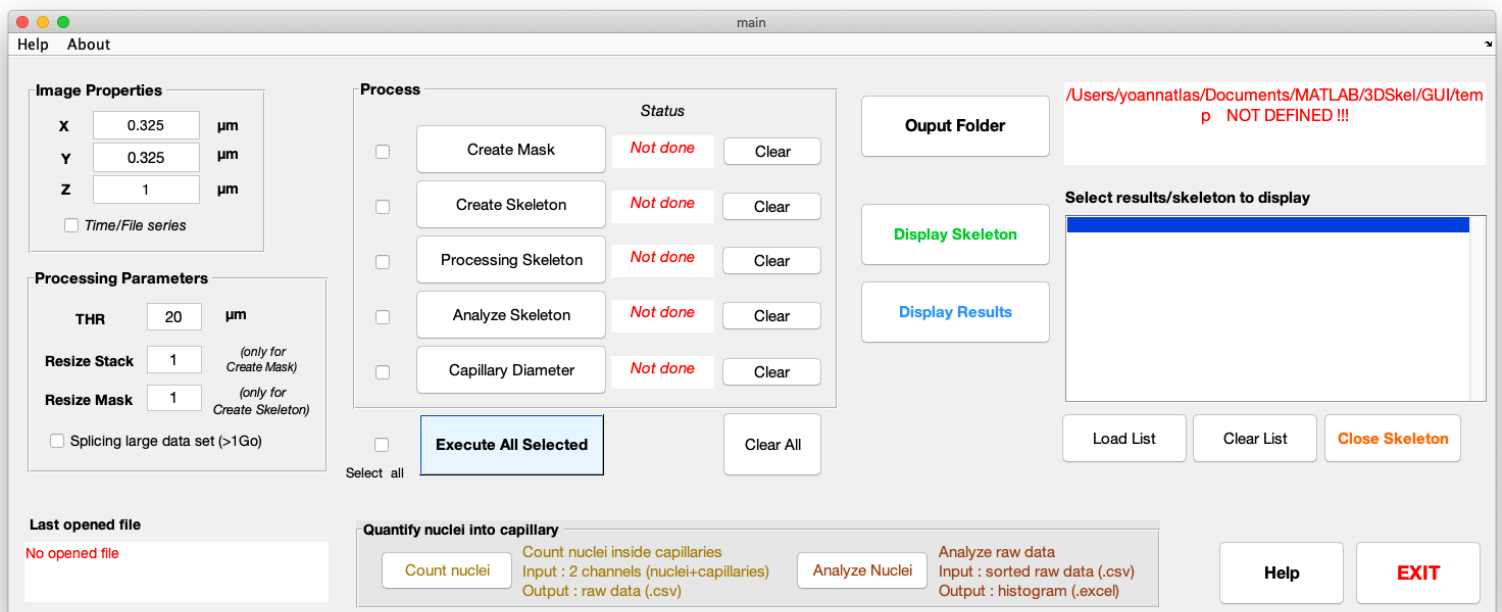


1. Overview

- A quoi ça sert ?

3DSkel permet de générer et quantifier un squelette à partir d'une image de fluorescence ou d'un masque binaire. 3DSkel traite des images uniques ou en batch, enregistre les résultats sous forme d'une feuille de calcul .xlsx et .csv ainsi que en format .mat et .fig (formats propriétaires MATLAB).

- Fenêtre principale



- Compatibilité

3DSkel a été développé sous MATLAB et utilise des fonctions de [ImageJ](#). La communication MATLAB ↔ ImageJ est assurée par le plugin [MIJI](#) (D. Sage, D. Prodanov, J-Y. Tinevez and J. Schindelin, *MIJ: Making Interoperability Between ImageJ and Matlab Possible*, 2012). Le logiciel [Fiji](#) v1.51i (ImageJ+plugins) est incorporé à 3DSkel en mode « no JREs » mais utilisant des plugins avec Java 1.8.0_152. Il est donc nécessaire d'installer Java 8 avec le système d'exploitation. Sur Windows, la version de Java utilisée est celle utilisée par MATLAB. 3DSkel a été testé sous la version MATLAB R2011a à R2018b. Sur Mac, les versions inférieures à R2018 utilisent Java 7. Les versions précédentes pourraient donc ne pas fonctionner à cause d'un problème de compatibilité Java.

3DSkel a été développé pour fonctionner sous Windows et sous Mac. La fonction `xlswrite` n'étant pas disponible sous Mac, celle-ci est remplacée par la fonction `xlwrite` développée par A. Zegher.

2. Image Properties

Les propriétés de l'image prennent en compte la taille d'un voxel en μm^3 . Par défaut la taille d'un voxel est de $0.325 \times 0.325 \times 1 \mu\text{m}^3$. La taille d'un pixel est importante pour la création d'un masque et l'analyse des résultats.

L'analyse en batch est possible. L'activation du mode batch (*Time/File series*) permet de traiter automatiquement tous les stacks contenus dans le même dossier.

3. Processing Parameters

THR indique le seuil de longueur de branche à appliquer lors du traitement du squelette. Toutes les branches, reliées à une ou deux extrémités et dont la longueur est en dessous du seuil, sont éliminées. La visualisation du squelette rend compte du seuillage grâce à un code couleur (rouge = squelette final, bleu = branches éliminées).

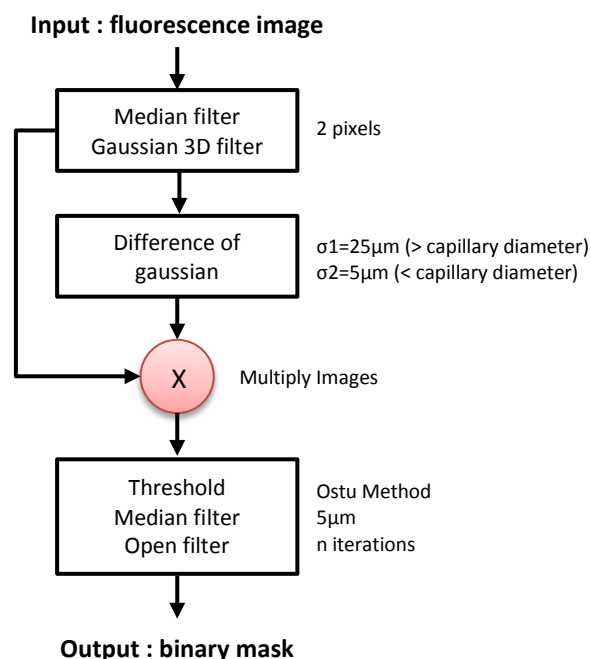
Par défaut, *THR* = 20 μm .

4. Functions

Principal functions

- **Create Mask**

La fonction Create Mask permet de générer un masque binaire à partir d'une image de fluorescence. L'image ne doit contenir qu'un seul canal correspondant aux capillaires. Les différentes étapes de création du masque sont détaillées sur le schéma ci-dessous.



- **Create Skeleton**

La fonction *Create Skeleton* transforme un masque binaire en squelette. Cette fonction n'intègre pas d'étape de filtration du masque. La squelettisation est effectuée par ImageJ avec le plugin [Skeletonize 2D/3D](#).

Le masque filtré est enregistré dans le dossier de sortie sous **MASK_PROCESS_xx.mat**

Le squelette brut est enregistré dans le dossier de sortie sous **SKEL_xx.mat**

- **Processing Skeleton**

La fonction *Processing Skeleton* applique le paramètre THR sur le squelette et retourne en sortie un squelette nettoyé des branches indésirables c.a.d. en dessous du seuil indiqué. La fonction utilise l'algorithme [skel2graph3D](#) développé par P. Kollmannsberger.

Lorsqu'une seule image est traitée, le squelette avant/après est affiché dans une fenêtre supplémentaire. La visualisation du squelette rend compte du seuillage grâce à un code couleur (rouge = squelette final, bleu = branches éliminées). La visualisation automatique est désactivée en mode batch.

Le squelette traité est enregistré dans le dossier de sortie sous **SKEL_PROCESS_xx.mat**

La visualisation 3D du squelette est enregistrée dans le dossier de sortie sous **xx.fig**

- **Analyze Skeleton**

La fonction *Analyze Skeleton* analyse et quantifie le squelette obtenu précédemment ou importé. Lorsqu'une seule image est traitée, les résultats sont affichés dans une autre fenêtre (DisplayResultsTable). L'affichage automatique des résultats est désactivé en mode batch.

Les résultats sont sauvegardés automatiquement dans le dossier de sortie sélectionné. Par défaut, les résultats sont sauvegardés dans un dossier temporaire.

Les résultats sont enregistrés dans le dossier de sortie sous **Results_xx.xlsx**.

Les fichiers **Results_xx.csv** et **BranchInfo.csv** ne servent qu'à la lecture des résultats sous MATLAB. **Attention à bien les conserver pour pouvoir relire les résultats sous 3DSkel !**

- **Capillary Diameter**

La fonction *Capillary Diameter* calcule le diamètre moyen de tout le réseau. La distance map est calculée à partir du masque par ImageJ avec le plugin [Local Thickness \(masked, calibrated, silent\)](#). Le diamètre est obtenu par superposition de la distance map et du squelette sur des projections tous les 20µm.

- **Execute All Selected**

La fonction *Execute All Selected* permet d'exécuter à la suite les différentes fonctions sélectionnées. L'exécution se fait toujours de haut en bas.

Le traitement par batch n'est activé que par cette fonction.

Secondary functions

- **Output Folder**

La fonction *Output Folder* sélectionne le dossier de sortie pour l'enregistrement des résultats. Si aucun dossier n'est sélectionné, les résultats ne seront pas enregistrés.

- **Display/Close Skeleton**

La fonction *Display Skeleton* affiche le squelette 3D dans une nouvelle fenêtre MATLAB. Le squelette affiché correspond au nom sélectionné dans la liste de résultats. Si aucun squelette n'a été généré précédemment, il est possible de sélectionner une figure de squelette déjà enregistré (xx.fig).

La fonction *Close Skeleton* ferme les fenêtres de visualisation du squelette en 3D.

- **Display Results**

La fonction *Display Results* affiche les résultats dans la fenêtre DisplayResultsTable.

- **Load/Clear List**

La liste permet de sélectionner les résultats à afficher si une ou plusieurs images ont été traitées. Chaque nouvelle image traitée s'ajoute à la liste. Un double-clic ouvre les résultats correspondants.

La fonction *Load List* importe une liste de résultats à partir des fichiers .xlsx, .csv et .fig. Les résultats peuvent ensuite être affichés.

La fonction *Clear List* efface la liste des images traitées.

- **Clear All**

La fonction *Clear All* supprime toutes les données temporaires.

- **Help**

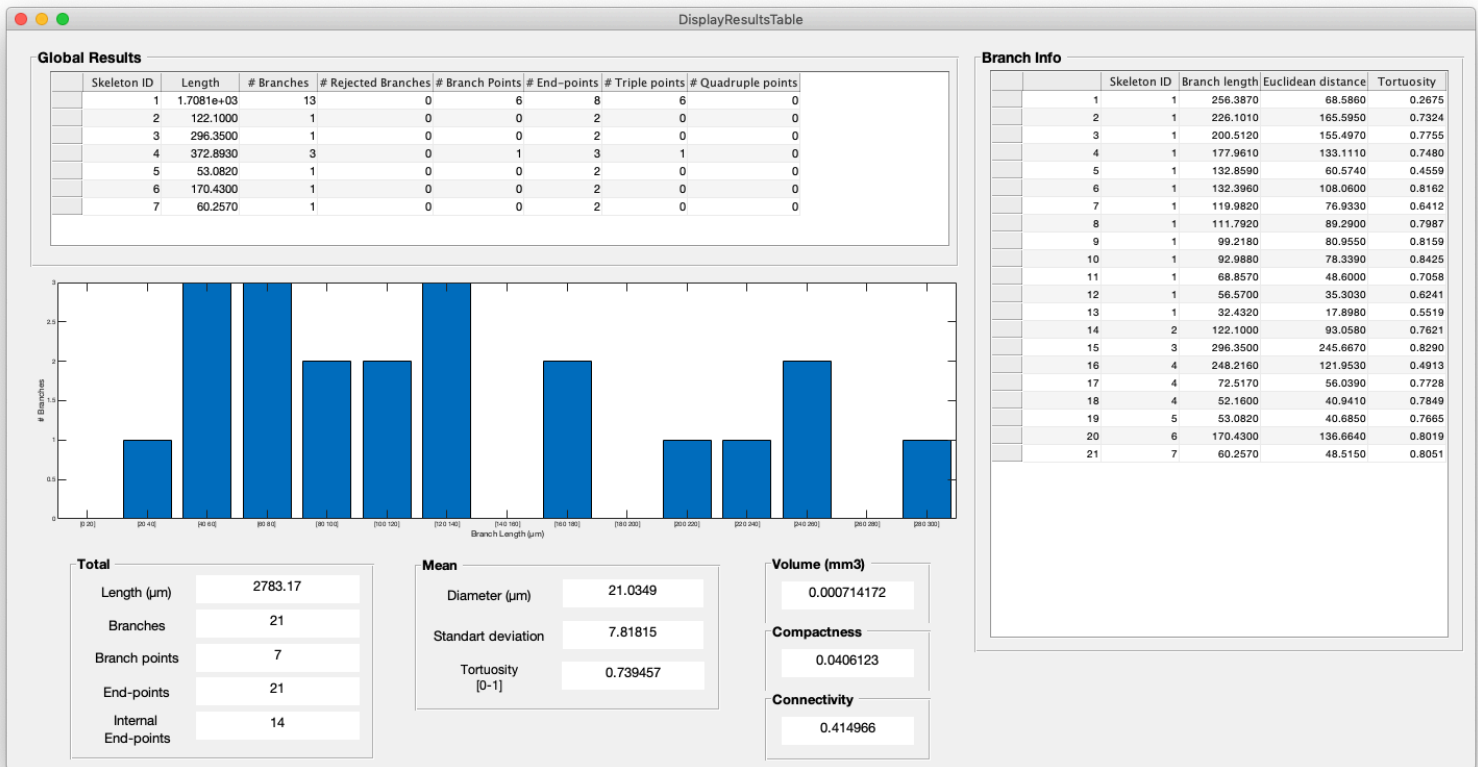
La fonction *Help* affiche la documentation du logiciel 3DSkel.

- **Exit**

La fonction *Exit* quitte le logiciel 3DSkel. Les données non sauvegardées seront effacées au prochain lancement de 3DSkel.

5. Results Table

Les résultats sont résumés dans la fenêtre suivante :



L'analyse du squelette est quantifiée de la façon suivante :

Global results /Branch info : tableaux regroupant les informations de chaque squelette et de chaque branche.

Histogram : nombre de branches triées par intervalles de longueurs de 20µm.

Length : somme de la longueur de toutes les branches du squelette après seuillage par THR (en µm).

Branches : nombre total de branches > THR.

Branch points : nombre total de points de branchement réels (les doublets pour les branches < THR sont supprimés).

End-points : nombre total de points d'arrêt.

Internal End-points : nombre total de points d'arrêt internes dans le stack (les points avec une marge de 15µm du bord du stack sont supprimés).

Nb skeleton : nombre total de squelettes indépendants.

Diameter : diamètre moyen de tout le réseau calculé à partir de projections tous les 20µm.

Standart deviation : écart-type du diamètre moyen.

Tortuosity : valeur permettant d'évaluer le caractère rectiligne du réseau (ratio distance euclidienne/longueur réel = [0-1]).

Volume : volume occupé par le réseau (en mm3).

Compactness : ratio volume occupé/volume total du stack.

Connectivity : indicateur global de connexité pondéré, adapté de la théorie des graphes

$$connectivity = \frac{\sum_{i=1}^{nb \ skeleton} B_i^2}{B_{total}^2}$$

avec B_i : nb de branches du squelette i

B_{total} : nb total de branches.

Troubleshooting

Le plugin *Local Thickness* (ImageJ) ne fonctionne pas si l'exécution de Fiji ne se fait pas en mode « no JREs ». Cela cause un problème de compatibilité de version Java entre MATLAB et Fiji.

La gestion de la mémoire sous Java (Java Heap Size Memory) est déterminée par MATLAB. Le maximum alloué par MATLAB est de 16Go. Pour contourner la limitation, un fichier *java.opts* doit être ajouté dans le dossier de démarrage de MATLAB (matlabroot/bin/arch folder/java.opts). Si des problèmes de mémoire apparaissent, vérifier que le fichier *java.opts* est bien ajouté au bon endroit. Le contenu du fichier est le suivant :

- **Xms2048m** **Initial Heap Size**
 - **Xmx32768m** **Max Heap Size**

Le plugin MIJI appelle une version de Fiji sans fermeture automatique des images. Les fichiers ouverts par MIJI doivent être impérativement fermés à la fin de leur utilisation sous peine d'une erreur Java *java.lang.OutOfMemoryError: Permgen Space*. Le vidage du cache de Fiji est activé par la macro *garbagecollector.ijm* à la fin de chaque utilisation.

Les fichiers sont découpés en ROI avant traitement sous Fiji. Ceci concerne le filtrage et la création du squelette. L'analyse du squelette ne peut être faite à partir de ROI. Pour de très gros fichiers (>Go) la fonction *Analyze Skeleton* (ImageJ) peut causer une erreur mémoire. Le redémarrage de MATLAB est alors nécessaire.

6. Support

Yoann Atlas

yoann.atlas@college-de-france.fr

Last version : 3DSkel v30