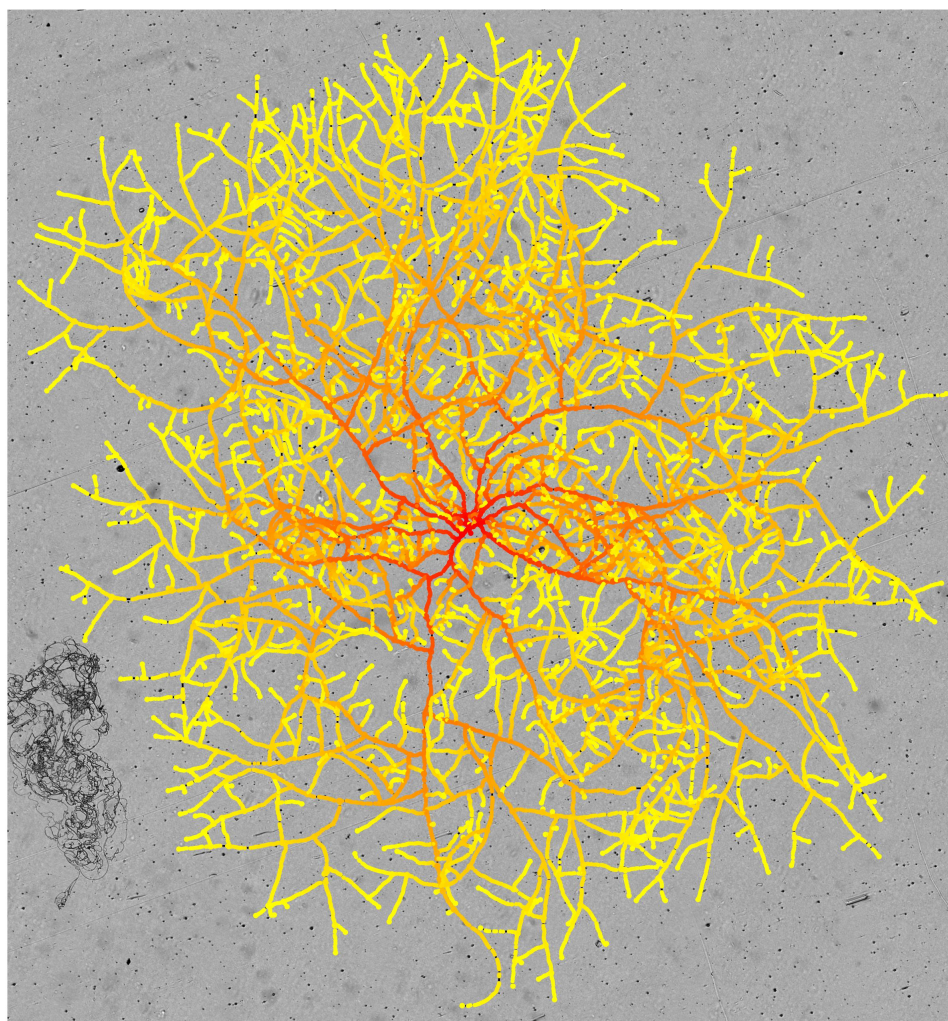


Podospora anserina : sa vie, son oeuvre, ses hyphes

Documentation utilisateur



Juillet 2023

Table des matières

1	Description de la fourniture	1
1.1	Aperçu des fichiers fournis	1
1.2	Liste des fonctions supplémentaires de net_utilities.py	2
2	Installation sur une nouvelle machine	3
2.1	Dépendances Python	3
2.2	Obtention de C_net_functions.so	3
2.3	Caméra : tiscamera	3
2.4	Platine : PIPython et PI Terminal	4
3	Se simplifier la vie avec un terminal	5
4	Acquisition des panoramas	6
4.1	Mise en place de l'acquisition	6
4.2	Démarrage des appareils	8
4.3	Lancement de l'acquisition	8
4.4	Suivi de la température et de l'humidité	11
5	Pré-traitement des panoramas	12
5.1	Reconstitution (stitching)	12
5.2	Binarisation	13
5.3	Nettoyage	14
6	Traitement des panoramas	15
6.1	Vectorisation - Superposition - Tracking	15
6.2	Vectorisation	16
6.3	Superposition	17
6.4	Tracking des noeuds	17
6.5	Tracking des noeuds	17
7	Conversion des anciens graphes	18
8	Liste des expériences	19

1 Description de la fourniture

1.1 Aperçu des fichiers fournis

Le dossier `Hyphes_2019` contient les dossiers suivants :

Codes qui contient tous les scripts nécessaires au traitement des panoramas.

Installation qui contient les fichiers nécessaires à l'installation des dépendances spécifiques.

Sauvegardes qui contient une sauvegarde zippée de ces scripts : un zip pour le code initial de Jana Lasser, un zip contenant les nouveaux codes de traitement et un zip pour l'application `Panoramyphes`.

Plus précisément, le dossier **Codes** contient les éléments suivants :

Panoramyphes un dossier qui contient les éléments nécessaires à l'exécution de l'application `Panoramyphes`.

StitchingMacro un dossier qui contient les macro `ImageJ` permettant la reconstitution et la binarisation des panoramas.

Dependencies.txt Ce fichier texte fait la liste de tous les modules Python nécessaires au bon fonctionnement des scripts et précise quelles versions ont été testées comme compatibles. Il précise également de quelle manière les modules sont installables.

C_net_functions.cpython-37m-x86_64-linux-gnu.so Un fichier python shared object, il est dépendant du système d'exploitation et doit être recompilé en cas de changement d'OS (voir Obtention de `C_net_functions.so`). Il est utilisé dans `net_utilities` pour des calculs un peu complexes qui sont plus efficacement exécutés en C (enfin c'est ce que j'ai compris...).

Graphs_conversion_part1_(python2).py Ce script en Python 2 sert à lire des graphes créés en Python 2 et à en sauvegarder une version qui est ensuite lisible en Python 3.

Graphs_conversion_part2_(python3).py Ce script en Python 3 permet de lire les objets créés par `Graphs_conversion_part1` et de les sauvegarder sous forme de graphes gpickle compatibles avec Python 3 et la version 2.3 de `NetworkX`.

net_utilities.py Ce module en Python 3 contient :

- une partie des fonctions définies par Jana Lasser dans son module `net_helpers`, mais adaptées pour fonctionner sous Python 3. Ces fonctions sont appelées par `Vectorisation.py` (et donc indirectement par `VST`) ;
- de nouvelles fonctions utilisées dans `Graphs_conversion_part2`, `Vectorisation`, `Superposition` et `TrackNodes` (et donc indirectement appelées par `VST`) ;
- de nouvelles fonctions appelées nulle part mais qui pourrait servir pour écrire des variantes des scripts existants (cf liste un peu plus bas dans Liste des fonctions supplémentaires de `net_utilities`).

Superposition.py Ce script en Python 3 permet de superposer les résultats d’une vectorisation et les panoramas (en niveau de gris ou binarisés).

TrackNode.py Ce script en Python 3 permet d’identifier les noeuds du réseau d’un panorama à un autre (et donc de faire un tracking de ceux-ci). Les identifiants des noeuds sont ajoutés aux graphes `gpickle`.

Vectorisation.py Ce script en Python 3 permet de vectoriser le réseau présent dans les panoramas. Il crée des fichiers `gpickle` (un par panorama) contenant toutes les informations du réseau.

VST.py Ce script 3 en 1 permet de lancer de manière séquentielle une vectorisation, une superposition et un tracking des noeuds sur un stack `tiff` de panoramas.

1.2 Liste des fonctions supplémentaires de `net_utilities.py`

Ces fonctions ne sont actuellement utilisées nulle part mais peuvent simplifier l’écriture d’un nouveau script. Pour une description plus détaillée, consulter les docstrings présentes dans le fichier `net_utilities.py`.

checkExtension vérifie si l’extension d’un fichier est bien celle attendue.

createNodesCSVForTracking crée et sauvegarde un fichier `csv` des coordonnées des noeuds depuis une liste de graphes.

drawNodesRandomColors dessine les noeuds d’un graphe sur une image.

extractDataForCSV extrait des `DataFrames` d’un graphe (utile pour la création ultérieure d’un fichier `csv`).

printAndUpdateLog met à jour un string de log et l’affiche (optionnel) dans la console.

preloadFiles parcourt un dossier et renvoie une liste ordonnée de tous les fichiers qui correspondent à l’extension voulue.

removeColumns supprime les colonnes spécifiées d’une `DataFrame`.

removeColumnsCSV supprime les colonnes spécifiées d’un fichier `csv`.

unstackTif sauvegarde séparément chaque image d’un stack `tiff` dans le format voulu.

writeLogAndExit met à jour un string de log, l’écrit dans un fichier puis ferme le programme en cours.

2 Installation sur une nouvelle machine

La liste des dépendances n'est pas exhaustive. Chaque machine étant différente, il est possible que des éléments supplémentaires soient nécessaires au fonctionnement des outils. Ces dépendances manquantes devront être traitées au cas par cas.

2.1 Dépendances Python

Toutes les dépendances sont installables via `pip` ou `pip3`. Certaines sont également disponibles via le navigateur **Anaconda** ou `conda-forge`.

Liste des imports Python :

<code>matplotlib</code>	<code>meshpy</code>	<code>networkx</code>	<code>natsort</code>
<code>numpy</code>	<code>opencv-python (cv2)</code>	<code>pause</code>	<code>pandas</code>
<code>pillow</code>	<code>pyserial (serial)</code>	<code>scipy</code>	<code>scikit-image (skimage)</code>
<code>shapely</code>	<code>tifffile</code>	<code>tqdm</code>	<code>trackpy</code>

2.2 Obtention de `C_net_functions.so`

Le code source et la procédure permettant de compiler le fichier `C_net_functions.so` en fonction de l'architecture proviennent du GitHub de l'auteur, Jana Lasser :

https://github.com/JanaLasser/network_extraction

1. Ouvrir le dossier `HypheS_2019 > Installation > net`.
2. Faire un clic droit dans la fenêtre du dossier et choisir `Ouvrir dans un terminal`.
3. Saisir la commande `python3 setup_C_net_functions.py build_ext --inplace`. Si cette commande ne fonctionne pas, essayer en remplaçant `python3` par `python`.
4. Deux fichiers sont normalement créés :
 - `C_net_functions.so` qui est à conserver ;
 - `C_net_functions.c` qui peut être supprimé.

2.3 Caméra : `tiscamera`

La dernière version du code source de ce module Python est disponible à l'adresse suivante :

<https://github.com/TheImagingSource/tiscamera>

La version de l'été 2019 se trouve dans `HypheS_2019 > Installation > tiscamera-master`. Pour l'installation se référer au fichier `README.md`.

2.4 Platine : PIPython et PI Terminal

2.4.1 PIPython

1. Ouvrir le dossier `Hyphes_2019 => Installation => PIPython-2.1.1.2` (ou une version plus récente si elle existe).
2. Faire un clic droit dans la fenêtre du dossier et choisir `Ouvrir dans un terminal`.
3. Saisir la commande `python3 setup.py install`. Si cette commande ne fonctionne pas, essayer en remplaçant `python3` par `python`.

Cette procédure se trouve également dans le fichier `readme.rst` du dossier `PIPython-2.1.1.2`. S'y référer pour plus d'informations.

2.4.2 PI Terminal

1. Ouvrir le dossier `Hyphes_2019 => Installation => PI_C663` (ou une version plus récente si elle existe).
2. Faire un clic droit dans la fenêtre du dossier et choisir `Ouvrir dans un terminal`.
3. Se connecter en temps que superutilisateur avec la commande `su`.
4. Saisir ensuite la commande `./INSTALL`. Si cette commande ne fonctionne pas, essayer en remplaçant `python3` par `python`.

Cette procédure se trouve également dans le fichier `README` du dossier `PI_C663`. S'y référer pour plus d'informations. Il est également possible qu'il soit nécessaire d'installer certaines des dépendances contenues dans le dossier.

2.4.3 Reconstitution des panoramas : stitchingMacro

La dernière version des macros de stitching des panoramas est disponible à l'adresse suivante :

<https://github.com/sebherbert/stitchingMacro>

Ces macros nécessitent l'installation d'ImageJ ou de Fiji pour être exécutées.

3 Se simplifier la vie avec un terminal

- Les flèche directionnelles (haut et bas) permettent de naviguer dans l'historique des commandes. Bien pratique pour ne pas avoir à retaper une commande que l'on a lancée juste avant !
- La touche **Tab** permet l'autocomplétion des noms de fichiers/dossiers. Par exemple, taper la commande `python Ve` puis appuyer sur **Tab** transformera automatiquement le `Ve` en `Vectorisation.py`.
- Dans un terminal, les raccourcis **Ctrl+C** et **Ctrl+V** permettant de faire des copier-coller ne fonctionnent pas. Il sont remplacés par **Ctrl+Alt+C** et **Ctrl+Alt+V**.

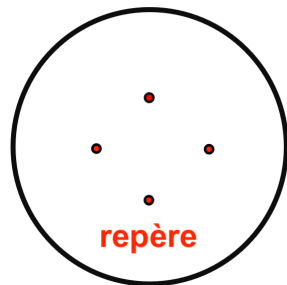
4 Acquisition des panoramas

L'acquisition des panoramas se fait sur l'ordinateur relié à l'enceinte thermostatée via l'application **Panoramyphes**. Utilisateur : dyco

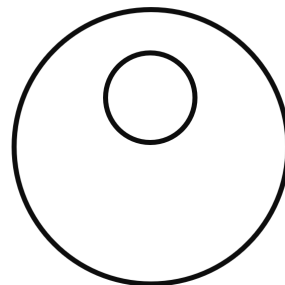
Mot de passe : lamarck

4.1 Mise en place de l'acquisition

Les différents couvercles :

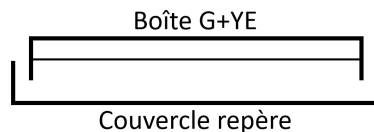


Couvercle repère

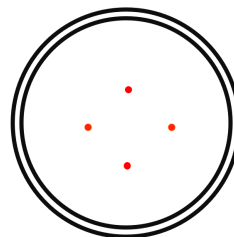


Couvercle acquisition

1. Faire projeter des ascospores sur couvercle d'agar à 27°C. Pour semer plus facilement ensuite, il peut être utile de placer le couvercle au moins 1h à 4°C pour solidifier l'agar.
2. Placer le couvercle REPÈRE sur une boîte de G + YE (yeast extract) et à l'aide d'un marqueur fin tracer 4 points au dos de la boîte comme sur le schéma ci-dessous. Éventuellement préparer 2 boîtes.



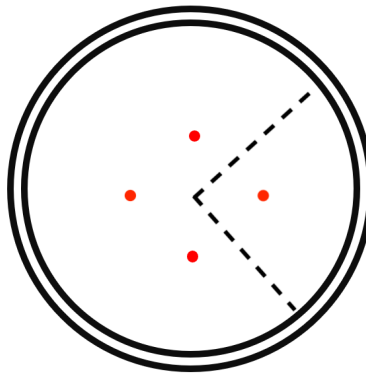
Vue de côté



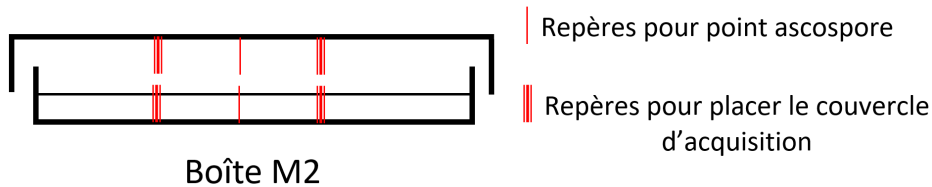
Vue du dessus

3. Préparer la boîte de G + YE :
 - Prendre une ou plusieurs feuilles de cellophane avec des gants et les mettre dans une boîte de Petri en verre avec de l'eau stérile. Faire bouillir au micro-onde ;
 - Déposer la feuille de cellophane sur la boîte de G + YE à l'aide d'une pince stérilisée en prenant soin d'éliminer toutes les bulles ;
 - Ouvrir la boîte et la mettre dans le four à UV (0,120 Joules) => Start), puis bien garder la boîte fermée pour éviter le dessèchement ;

- Récupérer le couvercle d'agar où les ascospores ont été projetées.
4. Stériliser très rapidement la pointe sous la flamme, puis semer délicatement une ascospore (prendre une grosse spore issue d'un asque de 4 ascospores) sur chacun des points en prenant soin de ne pas abîmer la cellophane. Garder la boîte bien à l'horizontal pour éviter le glissement des spores.
 5. Faire sécher la cellophane sous hotte 10min en enlevant le couvercle de la boîte, puis incuber 4 à 5h à 27°C (mettre les boîtes dans une feuille d'aluminium).
 6. À l'aide de la loupe, surveiller l'état de germination des ascospores. Dès que les ascospores ont germé, on peut soit conserver la boîte pour 2 jours maximum à 4°C, soit passer à l'étape suivante.
 7. À l'aide d'un scalpel stérilisé, découper le quart de boîte contenant l'ascospore choisie. Attention à bien couper la cellophane et à ne pas la déchirer.



8. À l'aide d'une pince stérilisée, prendre et déposer le morceau de cellophane où l'ascospore d'intérêt se trouve sur une boîte de milieu choisi pour l'expérience. Avec un scalpel, découper autour du quart de cellophane et enlever le surplus de milieu se trouvant dans la boîte. Vérifier à la loupe que l'ascospore est bien au centre et non juste au bord de la cellophane.
9. Sous la loupe, marquer un point sur le couvercle indiquant la position de l'ascospore (mettre le couvercle en butée). Marquer également un repère pour que l'ouverture du couvercle d'acquisition soit bien placée au dessus de la spore.



4.2 Démarrage des appareils

1. Allumer le cryostat pour le maintien de la température (2 boutons).
2. Allumer la lumière pour la caméra (gros boîtier beige `elc`, interrupteur noir sur le devant).
Le voltage doit être à environ 2,8 et le current à 0,1. Placer un filtre OD=3 sur la source lumineuse (car l'éclairage est trop puissant).
3. Vérifier que les deux contrôleurs de la platine (petits boîtiers gris **Mercury Step**) n'ont pas de voyant rouge allumé. Si un voyant rouge est allumé :
 - cliquer sur **Activités** en haut à gauche de l'écran ;
 - ouvrir l'application **PITerminal** en utilisant la barre de recherche ;
 - sélectionner le premier port USB et régler le baudrate sur 115200 puis cliquer sur **OK** ;
 - cliquer dans l'ordre sur les boutons F5 et F6 de l'application, puis sur **close** (en haut) et enfin sur **connect** ;
 - recommencer l'opération pour le second port USB jusqu'à **close** ;
 - fermer **PITerminal** en cliquant sur le deuxième **close** en bas de la fenêtre.

4.3 Lancement de l'acquisition

1. Ouvrir le dossier **Bureau > Hyphes_2019 > Codes > Panoramyphe**s.
2. Faire un clic droit dans la fenêtre du dossier et choisir **Ouvrir dans un terminal**.
3. Saisir la commande `python3 Panoramyphe.py` dans le terminal puis taper sur la touche **Entrée** pour lancer le programme.
4. Si le programme ne démarre pas et affiche une erreur (si les contrôleurs ont été débranchés par exemple), il faut déplacer les platines à l'aide de **PITerminal**, avant de lancer à nouveau **Panoramyphe.py**. Pour cela, reprendre l'étape 3. de la section 4.2 et entrer la commande `FNL 1` ou `FPL 1`.
5. Une fenêtre s'ouvre demandant si la platine va être utilisée. Répondre **Yes** sauf s'il est certain que celle-ci ne servira pas.
6. Déposer la boîte contenant l'ascospore sur le plateau, sous la caméra, de manière à voir quelque chose (pas nécessairement l'ascospore).
7. Dans les cas où la platine va être utilisée, il faut commencer par calibrer ses déplacements. Sélectionner **Platine Calibration** dans le menu **Settings** de la barre d'outils. La calibration ne prend que quelques secondes. Lorsque celle-ci est finie, le programme affiche une fenêtre d'information. Si la calibration est correcte, fermer la fenêtre et passer à l'étape suivante. Si elle est incorrecte, commencer par vérifier qu'il y a bien un objet sous la caméra (la calibration est impossible si la caméra ne voit rien ou un objet trop flou/indistinct).

Sinon réessayer la calibration. Si le problème persiste, relancer le programme. Si la calibration a été oubliée, n'importe quel mouvement de la platine entraînera une calibration automatique.

8. Il faut ensuite procéder au réglage de la caméra. Sélectionner **Camera Properties** dans le menu **Settings** de la barre d'outils.
 - Onglet **Exposure** : Mettre les 4 curseurs sur **OFF**. Régler **Brightness** sur 240 et **Gain** sur 0.
 - Onglet **Unknown** : Entrer la valeur de l'exposition dans **Exposure Time** (généralement aux alentours de 65).
 - Onglet **Image** : Mettre **Invert X** et **Invert Y** sur **ON**.
9. Placer l'ascospore dans le champ de la caméra. On peut s'aider d'une pointe fine et plate pour repérer la position de l'ascospore. Trois manières de faire :
 - en déplaçant directement la boîte manuellement ;
 - en déplaçant automatiquement la platine si les coordonnées de l'ascospore sont connues : aller dans **Auto** dans le menu **Motion** et saisir les coordonnées voulues (les coordonnées s'affichent en temps réel en bas à droite de l'application) ;
 - en déplaçant manuellement la platine : aller dans **Manual** dans le menu **Motion** et utiliser les flèches directionnelles comme indiqué. Si aucun déplacement n'est visible, soit la platine est en train de se calibrer, soit le focus n'est pas sur la fenêtre affichant la vue de la caméra.
10. Si les hyphes apparaissent transparentes à cause de l'humidité, attendre que celle-ci s'évapore et que les hyphes deviennent noires. Idem s'il y a des traces d'humidité sur la boîte.
11. Pour éviter le dessèchement lors d'une capture longue, retirer le couvercle et verser le milieu liquide presque à fleur de gélose mais pas trop pour éviter la submersion lors des déplacements (falcon + pipette si besoin). Mettre le couvercle d'acquisition en respectant le repère et vérifier la mise au point (tourner la tige pour régler la netteté en évitant de toucher la caméra). Enfin, bien fermer la porte de l'enceinte.
12. Lancer une acquisition en allant dans **Panoramas** dans le menu **Capture**. Les valeurs par défaut correspondent aux paramètres habituels. Sinon les modifier :
 - la longueur de déplacement en x et en y en mm (classiquement 10×10) ;
 - le taux de recouvrement en pourcentage (20%) ;
 - le temps de pause en secondes entre deux images (la valeur minimale possible est 2 secondes) ;
 - le temps de pause en secondes entre deux panoramas (classiquement 600s) ;
 - le nombre de panorama désiré.

Valider. Créer un nouveau dossier à l'emplacement souhaité avec l'icône en haut à droite (dossier avec un +) et le nommer comme désiré. Attention, ne pas mettre d'espace dans le

nom, on peut les remplacer par des underscores. Cliquer sur **Open** en haut à droite de la fenêtre.

13. La platine se recentre entre chaque panorama. En profiter pour vérifier la mise au point.
14. Pour arrêter l'acquisition, cliquer sur **End Capture** dans le menu **Capture**. L'application attendra la fin du panorama en cours avant de mettre fin à l'acquisition.

Il est également possible de mettre fin à l'acquisition en fermant directement le terminal. Cependant, il est vraiment préférable d'arrêter celle-ci lors d'une pause entre deux panoramas et non lors des prises de vue.

4.4 Suivi de la température et de l'humidité

Dans `Bureau > arduino > arduino-1.8.9`, ouvrir le fichier `arduino`.

Si le fichier `Prog.MMS` est ouvert :

1. téléverser le code avec la flèche de droite en haut à gauche de la fenêtre ;
2. ouvrir le moniteur série avec la loupe en haut à droite de la fenêtre.

S'il n'y a aucun fichier ouvert :

1. ouvrir le menu `Fichier` en haut à gauche de l'écran ;
2. sélectionner `Ouvrir` ;
3. ouvrir les dossiers `arduino` puis `arduino-1.8.9` ;
4. ouvrir le dossier `prog.MMS` puis sélectionner le fichier `prog.MMS` ;
5. téléverser le code avec la flèche de droite en haut à gauche de l'écran ;
6. ouvrir le moniteur série avec la loupe à droite de l'écran.

Pour sortir, fermer simplement les fenêtres.

5 Pré-traitement des panoramas

Les étapes de pré-traitement se font sur le serveur **hyphes**. Il faut donc transférer les images depuis l'ordinateur de manip vers le serveur.

Mot de passe serveur : **hyphes**

5.1 Reconstitution (stitching)

1. Dans `home > hyphes > Fiji.app` lancer Fiji en double-cliquant sur `ImageJ-linux64`.
Vérifier dans `Process > Binary > Options...` que la case `Black background` est cochée.
2. Ouvrir le dossier `home > hyphes > Hyphes_2019 > Codes > StitchingMacro`.
Glisser-déposer la macro `tiles2timeMosaic.ijm` dans la fenêtre de Fiji.
3. Dans la fenêtre `tiles2timeMosaic`, cliquer sur `Run` (en bas) puis renseigner les informations demandées :
 - choisir la première tuile `tile0.png` contenue dans le dossier `tiles` du répertoire d'acquisition ;
 - le nombre de tuiles en x et en y peut être obtenu en ouvrant le fichier `log.txt` du répertoire d'acquisition ;
 - pour `max number`, laisser la valeur par défaut (999). On peut choisir de traiter seulement certains panoramas, dans ce cas, entrer le numéro du premier et du dernier panorama (ex : 1 à 5).
 - 2 parcours "snake" existent pour le déplacement de la platine : soit en partant de la position "Down Left", soit "Up Right". Il faut indiquer quel parcours a été effectué durant l'acquisition (à vérifier dans le fichier `.ijm` dans le dossier d'un panorama). Avec le nouveau programme 2019, il faut choisir "Up Right".Cliquer sur `OK` pour lancer la macro.
4. Quand le programme a fini de tourner, il est écrit "Done stitching" dans la console de la fenêtre `tiles2timeMosaic`.
Fermer la fenêtre `tiles2timeMosaic`.

Deux dossiers ont été créés dans le répertoire de l'acquisition :

- `outputMovie` contenant `regMovie.tif` (un stack tif des panoramas) et `analysisParams.txt` ;
- `regMosaic` qui contient toutes les images du film au format tif.

5.2 Binarisation

1. Depuis le dossier `home > hyphes > Hyphes_2019 > Codes > StitchingMacro`, glisser déposer la macro `timeMosaic2bin.ijm` dans la fenêtre de Fiji.
2. Dans la fenêtre `timeMosaic2bin`, cliquer sur `Run` (en bas).

Pour `Select the registered movie`, choisir le film `regMovie.tif` précédemment créé et se trouvant dans `outputMovie`.

Cliquer sur `OK` pour lancer la macro.

3. Quand le programme a fini de tourner, il est écrit “Done Binarizing” dans une fenêtre `Log`. Fermer cette fenêtre.
4. Deux films sont affichés à l’écran : `bin_FFT_regMovie.tif` et `FFT_regMovie.tif`. Ils sont enregistrés dans le dossier `outputMovie`.

Pour chacun des films :

- inverser les couleurs en faisant `Ctrl+Maj+I` ;
- sélectionner `Yes` ;
- sauvegarder le film en faisant `Ctrl+S` ;
- sélectionner `Replace`.

5.3 Nettoyage

S'il n'y a pas d'impuretés, fermer les films et passer à la vectorisation. Sinon, il va falloir nettoyer. On effectue cette étape sur le film `bin_FFT_regMovie.tif`.

Après avoir sélectionné une zone à nettoyer, utiliser les options suivantes du menu **Edit** :

- **Clear** : supprime tout ce qui est dans la sélection ;
- **Clear Outside** : supprime tout ce qui est à l'extérieur de la sélection.

Attention, répondre **Yes** à la fenêtre de confirmation applique la suppression à toutes les images.

Pour supprimer plus rapidement une zone sur une seule image, utiliser la touche **Suppr** du clavier.

Une fois le nettoyage terminé, fermer les films (petite croix en haut à droite) puis sélectionner **Yes** puis **Replace**.

Astuce : Aller au dernier panorama et supprimer les impuretés sur les bords avec **Clear**, et choisir **Yes** pour appliquer à tous les panoramas. Pour les premiers panoramas, sélectionner l'ascospore et faire **Clear Outside**, puis répondre **No**.

Astuce : Sur Fiji, pour zoomer sur une image, faire `ctrl + molette` ; pour se déplacer sur une image, faire `espace + clic`.

6 Traitement des panoramas

Le traitement des panoramas se fait également depuis le serveur **hyphes**. Il consiste en une vectorisation du réseau, une superposition des images expérimentales et du résultat de la vectorisation, et un tracking des noeuds du réseau. Il est possible d'effectuer ces étapes en une seule fois (cf 6.1) ou séparément (cf 6.2, 6.3 et 6.4).

Dans tous les cas, il est nécessaire de commencer par :

1. Ouvrir le dossier `home > hyphes > Hyphes_2019 > Codes`.
2. Faire un clic droit dans la fenêtre du dossier et choisir **Ouvrir dans un terminal**.
3. Si l'expression **(base)** apparaît sur la ligne du terminal, ne rien faire. Sinon, saisir la commande `conda activate` dans le terminal (cela active l'environnement Python nécessaire à l'exécution de l'application).

Pour chacun des scripts, il est nécessaire de modifier les paramètres directement dans le script via un éditeur de texte. Pour cela, faire un clic droit sur le script concerné et sélectionner **Ouvrir avec Éditeur de texte**. Ne pas oublier de sauvegarder une fois les modifications apportées.

Pour lancer l'exécution d'un script (une fois les paramètres correctement réglés), saisir la commande `python nomDuScript.py` dans le terminal (exemple : `python Vectorisation.py`).

Durant l'exécution, des informations s'affichent dans le terminal et permettent de suivre le déroulement du programme. Un fichier texte de log (très originalement nommé `log.txt`) est également créé dans le dossier de destination ce qui permet une consultation ultérieure du déroulement du programme.

6.1 Vectorisation - Superposition - Tracking

Le script correspondant est nommé `VST.py`. Il permet d'effectuer séquentiellement une vectorisation, une superposition et enfin un tracking sur stack `tiff`.

Les paramètres à régler se trouvent ligne 24 à 100. Globalement, seul les chemins des fichiers et/ou dossiers sont à modifier. Le reste est déjà correctement réglé pour un usage courant.

Pour des explications plus précises sur chacun des paramètres, se référer au script de l'étape correspondante (vectorisation, superposition ou tracking).

Ouvrir `VST.py` avec l'éditeur de texte (on le trouve dans `Hyphes_2019` puis `codes`). Voici quelques indications concernant les modifications à apporter :

- **ligne 25** : mettre le chemin jusqu'au `FFT_regMovie.tif` (version non binarisée). Rq : on peut aussi mettre la version binarisée `bin_FFT_regMovie.tif`, on aura alors la superposition et le tracking sur les images binarisées
- **ligne 27** : mettre le chemin jusqu'au `bin_FFT_regMovie.tif`

- ligne 29 : pour créer un dossier où mettre les résultats
- ligne 30 : pour conserver une version `png` de chaque slice du stack binarisé
- ligne 31 : pour avoir des explications pendant que le programme tourne
- ligne 32 : donne encore plus d'explications (en général laisser `False`, sinon on a trop de détails)
- ligne 33 : inverse les images avant le traitement, peut aider s'il y a une erreur lors du traitement
- ligne 50 : le "pruning" permet de ne pas prendre en compte les toutes petites branches
- ligne 54 : pour changer le format des fichiers résultats
- ligne 67 : fait le stack
- ligne 68 : donne la vidéo (peut mettre un peu de temps à charger)
- ligne 69 : garder 3 en général
- ligne 72-81 : pour changer la couleur, la forme et la taille des branches, noeuds, apex
- ligne 89 : laisser `True` (crée un nouveau tableau Excel pour ranger les noeuds)
- ligne 91 : cherche dans un rayon de 10 (valeur modifiable) un noeud pour voir si c'est le même d'une image à l'autre
- ligne 92 : si un noeud disparaît pendant 3 images maximum, il sera conservé, mais au-delà de 3 images, il ne sera plus compté
- ligne 93 : si le programme sature avec 10 (cf ligne 91), il passe à 5
- ligne 97 : donne l'image du check
- ligne 99 : taille des ronds

Dans le terminal, écrire `python VST.py` et faire entrée. Un dossier `VST` apparaît dans le dossier de l'expérience en cours. Il contient la superposition et le tracking.

Rq : Avec Fiji, on peut passer si besoin d'un stack à des images séparées (ou l'inverse). Ouvrir le stack (ou les images), puis dans `Image > Stacks > Image to Stacks` ou `Stacks to Images`.

6.2 Vectorisation

Le script correspondant est nommé `Vectorisation.py`.

Il crée un graphe du réseau et sauvegarde celui-ci au format `gpickle`. En fonction des options, il peut également sauvegarder des images du réseau durant les étapes intermédiaires ainsi qu'une représentation graphique du graphe final.

Ce script s'applique aussi bien sur une image seule, sur un stack `tiff` ou même sur un dossier. Dans ce cas, toutes les images contenues dans le dossier seront traitées.

Pour fonctionner correctement, ce script nécessite des images binarisées au format `tif/tiff`, `png`, `bmp` ou `pgm`. Le format `jpg/jpeg` n'est pas supporté. Les autres formats n'ont pas été testés.

De plus, le réseau doit impérativement être blanc (valeur des pixels : 255) sur fond noir (valeur des pixels : 0). Dans le cas contraire, il est nécessaire d'activer l'option `invert` dans les paramètres.

Bien lire l'entête du script et de la fonction `init` (ligne 71) pour plus de détails.

6.3 Superposition

Le script correspondant est nommé `Superposition.py`.

Il dessine une représentation graphique des graphes sur les images correspondantes, permettant ainsi une vérification visuelle de la vectorisation.

Ce script nécessite donc les graphes `gpickle` ainsi que les stacks `tiff` d'images expérimentales ou binarisées. La superposition peut n'être faite que sur un seul graphe (une image simple en sortie) ou sur tous les graphes d'un dossier (un stack `tiff` ou une vidéo en sortie).

Bien lire l'entête du script et de la fonction `init` (ligne 31) pour plus de détails.

6.4 Tracking des noeuds

Le script correspondant est nommé `TrackNodes.py`.

Il effectue un tracking sur les noeuds des graphes `gpickle`, créant ainsi un lien entre tous les graphes. Les identifiants des noeuds sont ensuite réinjectés dans les fichiers `gpickle`.

Une option de vérification graphique est disponible. Elle dessine sur l'image fournie (a priori la dernière d'un stack) une croix pour chacun des noeuds de l'ensemble des graphes, avec une couleur différente pour chaque noeud identifié au cours du temps.

Bien lire l'entête du script et de la fonction `init` (ligne 41) pour plus de détails.

6.5 Tracking des noeuds

Pour récupérer le nombre d'apex (A), le nombre de noeuds (N) et la longueur totale (L) : `python3 get_all_nodes.py`. On obtient un fichier à 5 colonnes :

- numéro du pano
- nombre d'apex A
- nombre de noeuds N
- longueur totale L
- temps en secondes

7 Conversion des anciens graphes

La conversion des anciens graphes s'effectue à l'aide de deux scripts :

- `Graphs_conversion_part1_(python2).py`
- `Graphs_conversion_part2_(python3).py`

Cette conversion est nécessaire lorsque des graphes `gpickle` ont été obtenues à l'aide de l'ancien script de vectorisation, écrit sous Python 2. Il permet de convertir ces graphes en un format lisible en Python 3.

Après avoir réglé les paramètres des deux scripts à l'aide d'un éditeur de texte, suivre la procédure suivante :

1. Ouvrir le dossier `home > hyphes > Hyphes_2019 > Codes`.
2. Faire un clic droit dans la fenêtre du dossier et choisir `Ouvrir dans un terminal`.
3. Vérifier que l'expression `(base)` n'apparaît pas sur la ligne du terminal. Dans le cas contraire, utiliser la commande `conda deactivate` pour désactiver l'environnement `base`.
4. Saisir la commande `python Graphs_conversion_part1_(python2).py` dans le terminal pour lancer le premier script.
5. Une fois celui-ci terminé, activer l'environnement `base` avec la commande `conda activate`.
6. Saisir la commande `python Graphs_conversion_part2_(python3).py` dans le terminal pour lancer le second script.

8 Liste des expériences

Par défaut, la période d'acquisition est d'environ 20 minutes. La valeur précise peut être vérifiée dans le fichier `log.txt` pour chaque expérience.

Certaines expériences sont réalisées en triplicats (voir colonne N° réplicat), et d'autres sont indépendantes. Ces dernières peuvent ne pas être compatibles avec les autres expériences dans les mêmes conditions, notamment si les dates sont trop éloignées. Il peut en effet y avoir de légères modifications des produits utilisés pour les milieux de culture etc.

Condition	Manip	N° réplicat
blaa	bla	
	blabla	
bla		