

MASTER 2 BIOLOGIE COMPUTATIONNELLE PROMOTION
2024

Stage de fin d'étude

**Quantification de la transcription du VIH-1
dans le lymphocyte T**

Réalisé par : Raphaël Tranchot

Sous la direction de : Eugenia Basyuk (DR2 – CNRS)

Janvier 2024 - Juin 2024

Table des matières

1	Introduction	2
2	Etat de l'art	5
2.1	Détection des cellules	6
2.1.1	Cellpose	6
2.1.2	StarDist	6
2.2	Suivi des cellules	6
2.2.1	Ultrack	7
2.2.2	Deepcell	7
2.2.3	Napari	7
2.3	Détection et quantification des molécules	8
2.3.1	FishQuant	8
2.3.2	BigFishLive	9
2.4	Conclusions de l'Etat de l'art	17
2.4.1	Problèmes liés à l'utilisation de BigFishLive	17
2.4.2	Choix des ressources	19
3	Matériel et Méthodes	20
3.1	Matériels	20
3.1.1	Ressources	20
3.1.2	Développement préliminaires	20
3.2	Méthodes	21
3.2.1	Segmentation	21
3.2.2	Détection et quantification	24
4	Résultats et Discussion	27
4.1	Résultats	27

4.2 Discussions	28
4.2.1 Changement de paramètres	28
4.2.2 Application et Perspectives	30
5 Remerciement	32
6 Annexe	35
6.1 Jeu de données	35
6.2 Obtention des outlines	36
6.2.1 Pipeline	36

1 Introduction

Le VIH est un rétrovirus, dont le génome est composé d'ARN double brin [6]. Ce virus est responsable du Syndrome d'ImmunoDéficience Acquise (SIDA), maladie touchant près de 39 millions de personne dans le monde. Actuellement, il n'existe aucun moyen de guérir du SIDA. Cependant, il existe une trithérapie permettant de réduire la charge virale du VIH afin d'éviter l'apparition du SIDA, stade où différents symptômes associés à une immunodéficience acquise s'expriment et portent atteinte de façon significative à la santé et à l'espérance de vie du malade [2]

Malgré les progrès dans le domaine, une trithérapie reste contraignante pour le patient (traitement à vie, toxicité sur le long terme, etc.), et ne permet pas d'éradiquer le virus. En effet la trithérapie ne permet que de s'attaquer aux virus actifs et non aux virus latents incorporés dans le génome des cellules hôtes. Les virus latents sont invisibles pour le système immunitaire et ne peuvent pas être réprimés par les antiviraux. En cas d'arrêt de la trithérapie, l'infection reprendra par un mécanisme de transcription de l'ADN viraux en ARN qui mènera in fine à la production et la propagation de nouveaux virus dans l'organisme. La transcription est une étape clé dans la régulation de la latence virale.

La transcription du VIH étant stochastique, il y aura donc une transition aléatoire entre deux états possibles du promoteur *ON* et *OFF* qui vont mener respectivement

à :

- La transcription de l'ADN viral afin de poursuivre le cycle de réPLICATION du virus
- ou l'état de latence, état dans lequel il n'y a pas de transcription et où le virus sera indéTECTABLE pour le système immunitaire.

Le HIV Transcription Project de l'équipe Andevir du MFP étudie la transcription du VIH dans les cellules vivantes par microscopie. L'équipe cherche à comprendre comment le promoteur varie entre les états *ON* et *OFF*. Pour cela l'équipe a produit des clones de lymphocytes T avec le rapporteur VIH intégré dans différents sites dans le génome pour observer la transcription en fluorescence.

Pour visualiser et quantifier les molécules d'ARN dans les cellules vivantes l'équipe utilise MCP-StayGold (voir Figure 1.1). Cette technique utilise la propriété de la protéine MCP (MS2 Coat Protein) qui a une très forte affinité avec la séquence MS2 de l'ARN transcrit. Ce dernier possède un motif avec 128 répétition de MS2. MCP-StayGold va donc se fixer sur toute la longueur de ce motif et StayGold permettra de rendre visible l'ARNm en microscopie à fluorescence. (voir Figure 1.2)

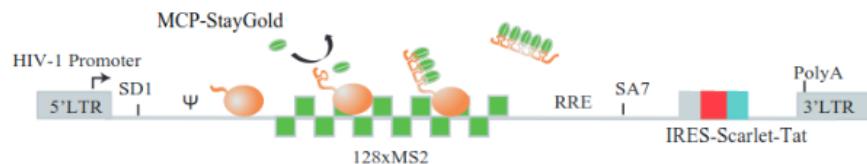


FIGURE 1.1 – Système expérimental

Schéma du rapporteur du VIH montrant le motif MS2 répété 128 fois et son interaction avec MCP lors de la transcription de l'ARNm.

Le rapporteur code pour Tat (Trans-Activator of Transcription), protéine qui augmente l'efficacité de la transcription virale. [9] Tat est fusionné avec une protéine fluorescente rouge, Scarlet.

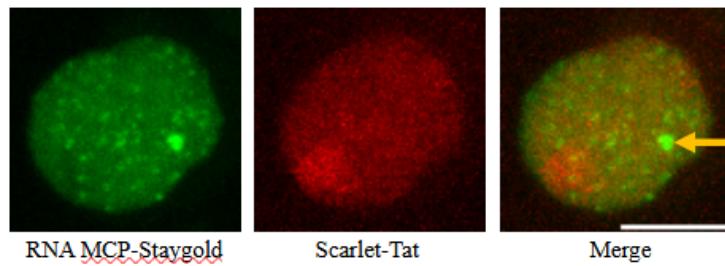


FIGURE 1.2 – Cellules avec le rapporteur VIH.

À gauche : marquage par MCP-StayGold les points sont les molécules uniques d'ARN (voir Figure 1.1), le gros point correspond au site de transcription (longueur d'onde : 488 nm). Au centre : marquage de noyau par Scarlet-Tat (activateur de la transcription) (longueur d'onde 562 nm). À droite : Superposition des deux images, le site de transcription est indiqué par une flèche.

Le jeu de données correspond à des stacks 3D d'images de cellules prises chaque minutes dont l'espacement en z est de 700 nm. (Voir Figure 1.3)

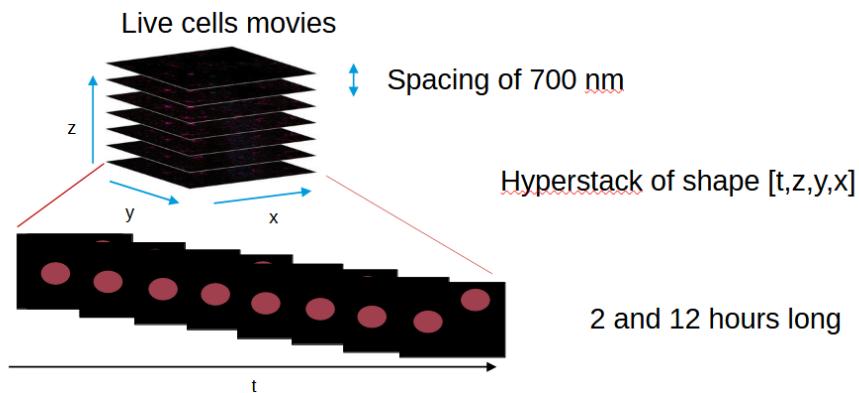


FIGURE 1.3 – Format des films

Le format des films utilisés correspond à un Hyperstack de forme $[t,z,y,x]$ avec t correspondant au temps en minute et z,y,x correspondant aux 3 dimensions en nanomètre

Ce qui en fait des films 3D où l'on suit un groupe de cellule en fonction du temps.

Pour analyser ces films, nous réaliserons une projection basée sur l'intensité maximale pour convertir des images 3D en 2D. Pour chaque pixel dans l'image projetée,

l'algorithme va parcourir toutes les tranches du stack en z et sélectionner la valeur la plus intense pour ce pixel. Une fois les films ramenés en 2D nous pourrons visualiser les cellules avec leurs déplacements et l'apparition/disparition des sites de transcription (Voir Figure 1.4).

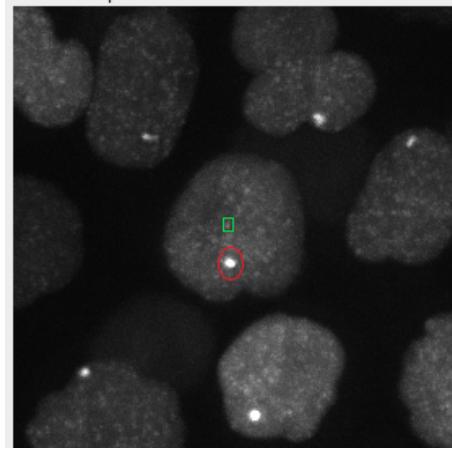


FIGURE 1.4 – Projection 2D d'un film à un instant t
En rouge : site de transcription. En vert : molécule unique d'ARN.

Pour étudier plus profondément le mécanisme de transcription du promoteur, il est nécessaire de quantifier le nombre de molécules d'ARNm naissantes présentes aux sites de transcription. Le but de mon stage consiste à établir un pipeline permettant de réaliser une analyse quantitative des molécules d'ARNm viraux aux sites de transcription, dans des clones de lymphocytes (cellule Jurkat) vivantes. Les résultats obtenus permettront d'établir un modèle du promoteur du VIH et d'élucider les mécanismes de transition entre l'état actif *ON* et non-actif *OFF*.

2 Etat de l'art

Le but étant de quantifier des molécules sur des films de cellules individuelles, plusieurs problématiques se posent :

- La détection des cellules

- Le suivi des cellules détectées
- La détection des molécules d'ARNm
- Et la quantification des molécules d'ARNm aux sites de transcription

2.1 Détection des cellules

La détection sera réalisée par segmentation. La segmentation consiste à diviser une image en segments, lesquels correspondent aux formes des cellules. [1]

Pour la segmentation différents packages Python existent tels que Cellpose [10] et StarDist [8].

2.1.1 Cellpose

Cellpose utilise un modèle généraliste de Deep Neural Network, pré-entraîné avec un large dataset de plus de 70000 objets. Il permet de segmenter des images 2D et 3D pour une grande diversité de type cellulaire. Différents paramètres sont adaptables tels que le diamètre des cellules ou le type d'élément à segmenter (noyau ou cytoplasme).

2.1.2 StarDist

StarDist repose sur un algorithme de type Convolutional Neural Network (CNN) et localise les cellules via des polygones. L'utilisation de polygones permet d'obtenir une segmentation plus précise comparé à l'utilisation de boîtes englobantes, notamment dans le cas des images avec une forte densité de cellule.

2.2 Suivi des cellules

Pour suivre la transcription de chaque cellule dans le temps nous devons les suivre malgré leurs déplacements et leurs changements de forme. Pour cela nous réaliserons une opération de tracking, qui permet d'attribuer à chaque cellule une étiquette unique au travers d'un film.

2.2.1 Ultrack

Ultrack [4] est un package Python qui permet de réaliser de la segmentation et du tracking de façon conjointe. Il est conçu pour être efficient sur de large datasets contenant des millions d'instances de segmentations. Il est possible de l'utiliser avec d'autres modèles de segmentation, ce qui lui permet de réaliser le tracking sur une grande diversité de type cellulaire.

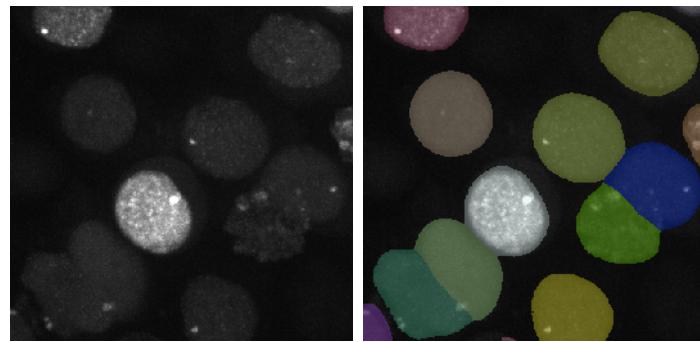
2.2.2 Deepcell

D'autres packages Python comme Deepcell [5] existent et sont utilisés sous forme de notebook Jupyter. Deepcell repose sur une méthode d'apprentissage supervisé avec un réseau de type CNN (Convolutional Neural Networks).

Ultrack et Deepcell utilisent les mêmes méthodes pour détecter et suivre les cellules, la différence réside dans leurs exécutions. Ultrack utilise entre autre Napari (qui est expliqué plus bas dans la section 2.2.3) et permet donc d'agir sur chaque frame du film de façon plus interactive que Deepcell qui travaille sur des films de cellule en format vidéo.

2.2.3 Napari

Napari [7] est un logiciel développé en Python et permet de visualiser et traiter des images biologiques. Il se base sur le principe de couches permettant de superposer les images ou des formes (point, traits, rectangles, etc.) à la manière de calque. Cette fonctionnalité est utile pour le suivi des cellules car on peut superposer les films avec les masques de segmentation/de tracking obtenus (voir Figure 2.1).



(a) Sans masque de seg-
mentation (b) Avec masque de seg-
mentation

FIGURE 2.1 – Extrait de film avec et sans couche qui contient les masques de seg-mentation

2.3 Détection et quantification des molécules

Pour détecter des molécules à l’intérieur de cellule de nombreuses ressources existent :

- Fishquant : détection de spots qui correspondent à des molécules d’ARN sur les images de smFISH, développés pour Matlab.
- Bigfish : détection de spots développés pour Python.
- Des outils recommandés et/ou conçus par des collaborateurs (BigFishLive).

2.3.1 FishQuant

Fishquant [3] développé par l’équipe de F.Mueller permet de réaliser la quantification de spots correspondant à des molécules uniques d’ARN qui sont présentes sur des images de cellule fixées. Cet outil à l’origine conçu en Matlab existe en Python sous le nom de Bigfish. FishQuant n’utilise pas des méthodes de deep learning mais utilise plutôt des méthodes de traitement d’images telles que la détection et la quantification de pics d’intensité.

2.3.2 BigFishLive

Développé par Rachel TOPNO, en PhD de bio-informatique à l'IGH (Institut de Génétique Humaine) de Montpellier. BigFishLive adapte les fonctions existantes de BigFish pour les films de cellules vivantes. Le pipeline consiste en deux grandes parties :

- La segmentation et le découpage d'images de cellule
- La détection des molécules uniques d'ARN et la quantification des molécules d'ARN aux sites de transcription

Le pipeline comprend plus précisément des étapes de :

- Projection des films dans le temps et segmentation afin d'obtenir le contour du déplacement des cellules.
- Découpage des cellules à partir des résultats de segmentation afin d'obtenir des films pour chaque cellule.
- Test de la fluctuation de l'intensité afin d'analyser le rapport signal/bruit de fond.

Puis des étapes relatives à la détection et quantification de molécules uniques d'ARN et de sites de transcription :

- Obtention d'un seuil pour la détection de molécules uniques.
- Construction d'un spot de référence.
- Tracking des sites de transcription et correction manuelle si nécessaire.
- Quantification des sites de transcription

Nous allons faire une revue non exhaustive des étapes de ce pipeline que nous jugeons les plus intéressantes.

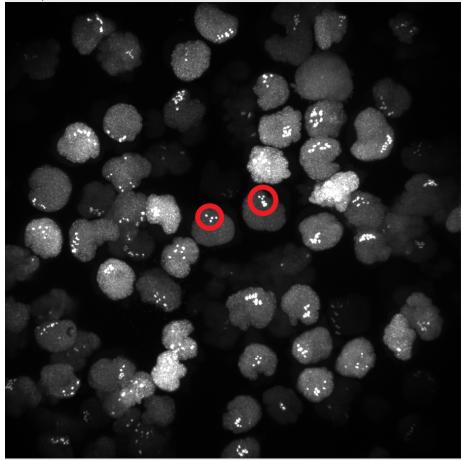
Segmentation et découpage

Projection des films dans le temps et segmentation :

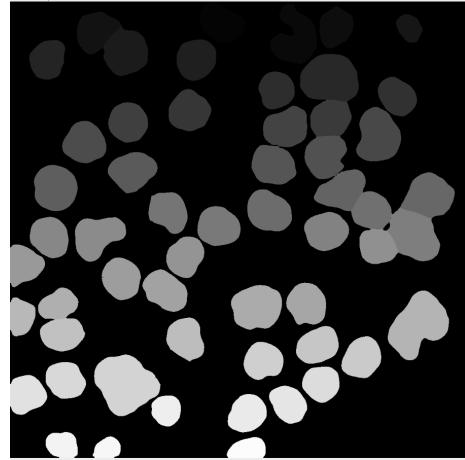
Il y aura une projection des films en fonction du temps (méthode de l'intensité maximale). La projection dans le temps consiste à superposer sur une même image tous les déplacements de cellules. Ainsi on aura sur une même image l'étendue du déplacement de chaque cellules. (voir Figure 2.2a)

Une fois la projection faite on va segmenter avec Cellpose. On va pour chaque

cellule détecter le contours de son déplacement sur tout le film. Le masque obtenu sera sauvegardé. (voir Figure 2.2b)



(a) Projection



(b) Masque de segmentation

FIGURE 2.2 – Projection d'un film dans le temps (a) et masque associé (b)

Sur la figure (a) on peut voir le déplacement de sites de transcription dans chaque cellule entouré en rouge. Les points multiples correspondent à la position des sites à différents moments du film.

Translation et découpage des cellules :

Cette étape consiste à utiliser les films et les masques obtenus pour découper des films individuels pour chaque cellule. Pour cela il y aura un calcul des bounding box et des centroides à utiliser pour découper et centrer chaque cellules à travers le film.(voir Figure 2.4).

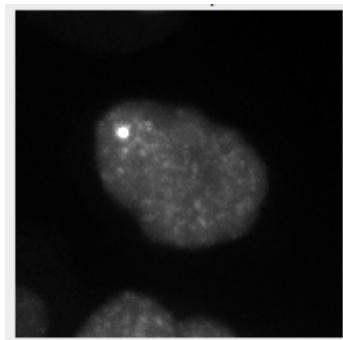


FIGURE 2.4 – Projection 2D d'un instant t du film d'une cellule découpée

Calcul de la fluctuation de l'intensité :

Cette étape consiste à calculer l'intensité moyenne à travers le film. Pour cela nous sélectionnons une cellule et une zone sans cellules correspondant au bruit de fond dans Napari(voir Figure 2.5).

Cette étape nous permettra de mesurer le photoblanchiment, si il existe dans le film (voir Figure 2.6).

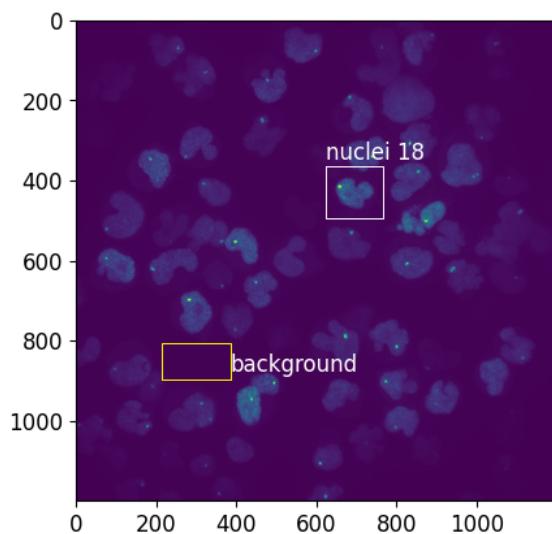


FIGURE 2.5 – Sélection d'une cellule et du background

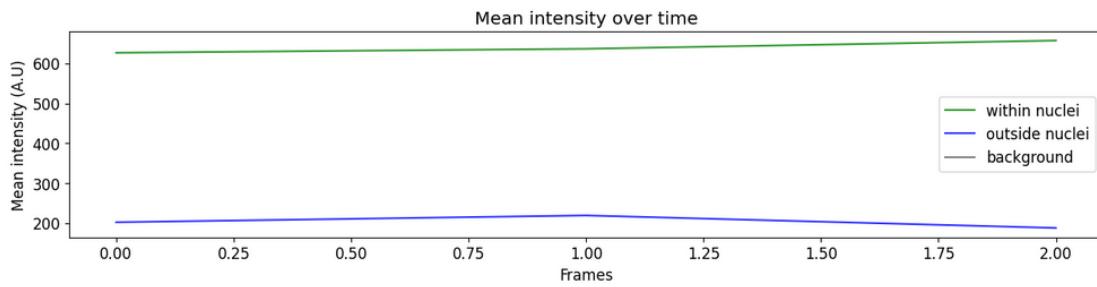


FIGURE 2.6 – Intensité moyenne en fonction du temps

Ici l'intensité moyenne du bruit de fond est très inférieure à celle en dehors du noyau

Détection et quantification

Obtention d'un seuil :

Une fois les films découpés, il faut déterminer le seuil pour chaque cellule, deux méthodes sont disponibles :

- La méthode par défaut de BigFish
- Et la méthode basée sur la différence des courbes normalisées

Ces seuils correspondent à une valeur d'intensité permettant de détecter molécules individuelles d'ARN et celles aux sites de transcription.

La méthode par défaut utilise la fonction `detection.detect_spots` qui prend en paramètre la taille du voxel et le rayon du spot en nanomètre et va estimer automatiquement une taille de kernel pour le filtrage LoG et une distance minimale entre deux spots que l'on souhaite pouvoir détecter séparément. Une approximation des rayons du spot en pixel se fera avec la fonction `detection.get_object_radius_pixel`.

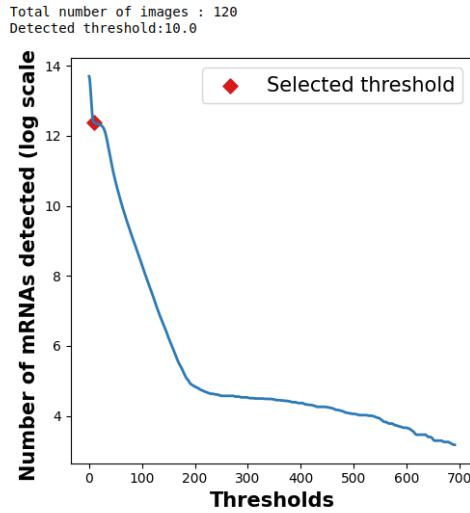


FIGURE 2.7 – Courbe des seuils obtenu avec la méthode par défaut de BigFish
Le graphe représente le nombre d'ARN détecté en fonction de l'intensité (échelle logarithmique). Ici le seuil = 10.

La méthode de la différence des courbes normalisées récupère des valeurs d'intensités des images afin de créer un coude qui sera utilisé automatiquement pour régler le seuil afin de détecter les spots.

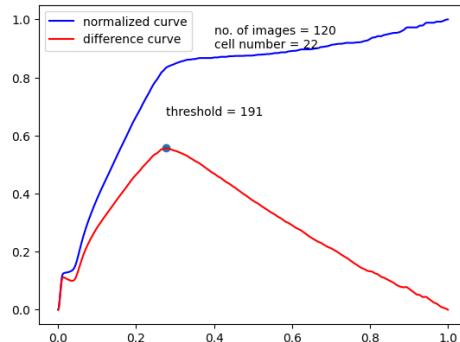


FIGURE 2.8 – Courbe normalisées des seuils obtenus
Pour un film de 2 heures le seuil est de 191

En fonction de la méthode, la valeur des seuils peut varier. Dans notre cas, nous travaillons avec des films ayant 2 types de structures à détecter :

- Les spots = molécules uniques d'ARN
- Les clusters = accumulation de molécules uniques dans un rayon, dans notre cas cela correspond généralement aux sites de transcription.

Pour choisir le seuil, nous utilisons généralement la valeur obtenue par la méthode des courbes normalisées que nous divisons par deux. Des ajustements peuvent être fait en fonction du film.

Construction d'un spot de référence :

Nous utilisons le seuil déterminé pour réaliser une première détection de tous les spots à travers le film, qui seront stockés dans une liste. Cette liste sera utilisée pour construire un spot de référence, qui correspond à l'image moyenne d'une molécule unique d'ARN à travers le film pour une cellule donnée.

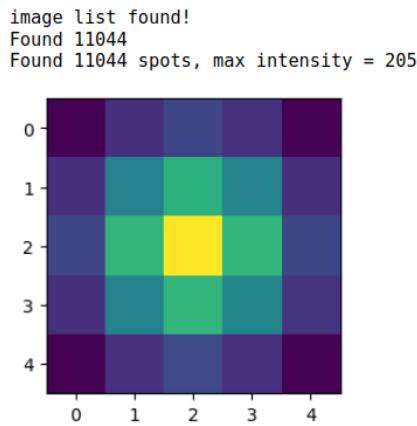


FIGURE 2.9 – Spot de référence d'une cellule

11044 spots ont été détectés à travers un film de 2 heures dans une cellule découpée pour une intensité maximale de 205

Détection des spots et clusters :

Avec le spot de référence, on peut procéder à la détection définitive des spots et la détection des clusters sur tout le film. On obtiendra 2 tableaux numpy correspondant à la liste des spots et des clusters en fonction du temps, avec leurs coordonnées en z, y et x. Pour les clusters, il y aura en plus le nombre de molécules par clusters et un index. [2.10](#)

```

clustersFrames[0]

array([[ 9, 127, 94, 2, 0],
       [ 12, 74, 86, 2, 1],
       [ 12, 109, 114, 2, 2],
       [ 12, 115, 99, 2, 3],
       [ 12, 141, 76, 2, 4],
       [ 13, 85, 81, 2, 5],
       [ 13, 131, 85, 2, 6],
       [ 13, 144, 69, 2, 7],
       [ 14, 119, 82, 3, 8],
       [ 14, 121, 57, 2, 9],
       [ 15, 117, 65, 2, 10],
       [ 15, 125, 64, 3, 11]])

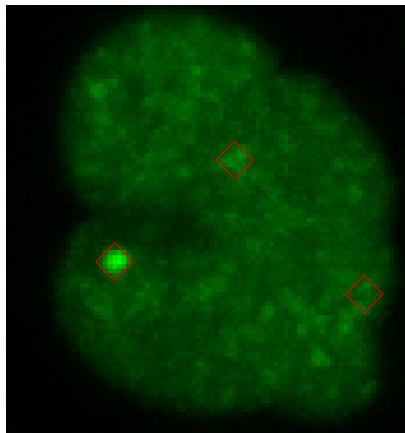
```

FIGURE 2.10 – Liste de clusters à l'instant 0

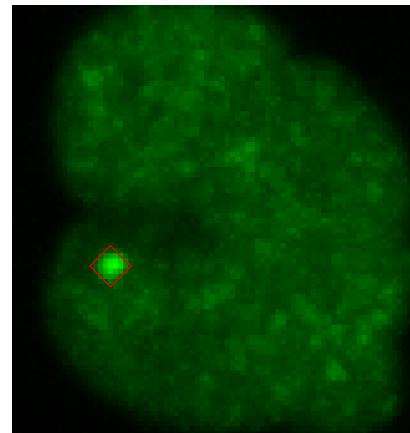
La liste a une forme de type [z,y,x,n,i], avec z,y,x correspondant aux coordonnées tridimensionnelles du cluster, n le nombre de molécule et i l'index du cluster dans la liste

La liste des clusters correspond à une détection préalable des clusters dans le film. Pour chaque instant on aura de multiples candidats que BigFish considère comme clusters.

Par la suite le reste du pipeline est censé permettre de trier ces clusters et de déterminer les vrais sites de transcription. [2.11](#)



(a) 3 Clusters candidats



(b) Cluster réel

FIGURE 2.11 – Exemple de cellule avec plusieurs clusters candidats dont seul un correspond au cluster réel

Les clusters (sites de transcription) sont encadrés par des carrés rouge

Choix des paramètres

Pour améliorer la détection des spots et des clusters de nombreux paramètres sont disponibles comme :

- Le nombre de spots pour considérer qu'une région est un cluster
- Le rayon dans lequel ses spots peuvent se trouver (entre 300 et 700 nm)
- Le seuil d'intensité pour détecter ces spots (dépend du seuil déterminé)
- des paramètres liés à différentes fonction

Parmi ces fonctions il y a notamment *buildReferenceSpotFromImages* :

- alpha : qui influence le nombre de spots par region candidate. Un alpha de 0.5 permet d'avoir un spot de référence correspondant au spot médian
- gamma : qui influence l'étape de filtration pour retirer le bruit de fond de l'image.

Dans le cas de la fonction *getSpotAndClusters* nous avons des facteurs de multiplication comme :

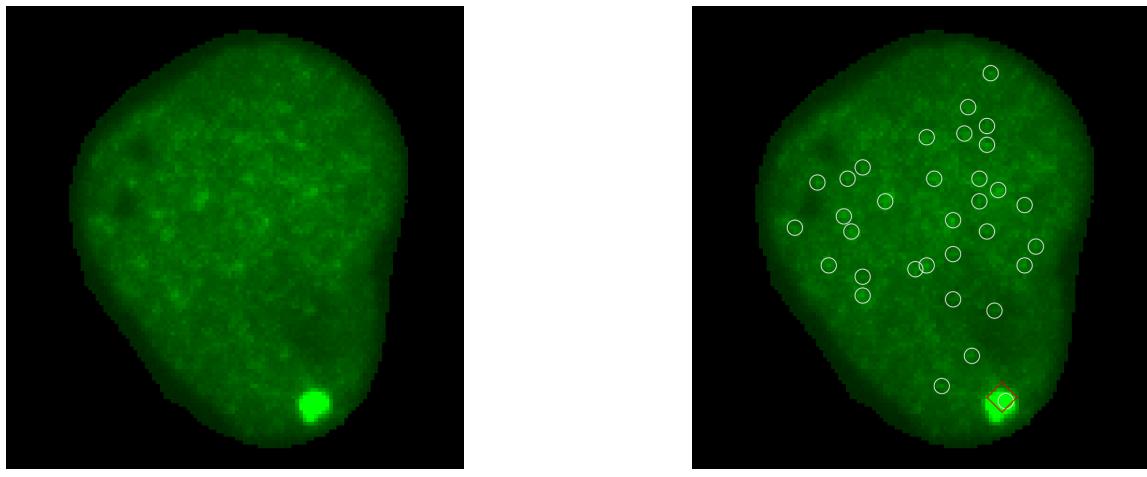
- beta est un facteur de multiplication pour le seuil d'intensité d'une région dense : $threshold = \beta * max(medianspot)$
- gamma, facteur de multiplication pour calculer la taille du noyau gaussien : $kernelsize = \frac{\gamma * spotradius}{voxelsize}$

Après avoir testé de plusieurs réglages nous avons choisi comme paramètres :

- $\alpha = 0.5$
- $\beta = 2$
- $\gamma(getSpotAndClusters) = 15$
- nombre de molécule = 2
- $rayon = 400nm$

Résultats

Ce script peut donner de bons résultats dans le cas de films courts ou les cellules ne se déplacent pas beaucoup, par exemple si les cellules sont adhérentes.



(a) Sans détection

(b) Avec détection

FIGURE 2.13 – Exemple de détection de molécules individuelles et de sites de transcription

Les molécules uniques sont entourées par des cercles et le site de transcription est encadrée par le carré rouge

2.4 Conclusions de l'Etat de l'art

2.4.1 Problèmes liés à l'utilisation de BigFishLive

Au cours du test de BigFishLive de nombreux problèmes ce sont posé dans le cas de nos films :

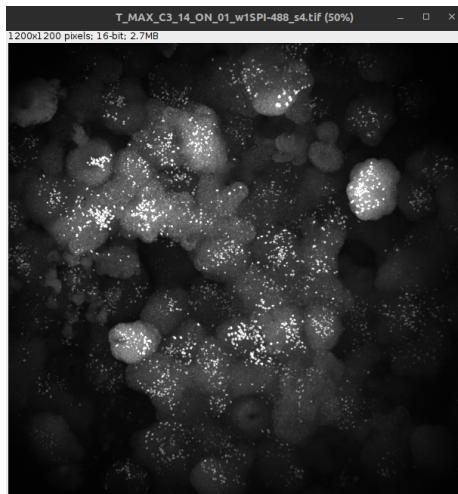
- Photoblanchiment
- Segmentation et découpage des cellules
- Isolement des cellules

Photoblanchiment

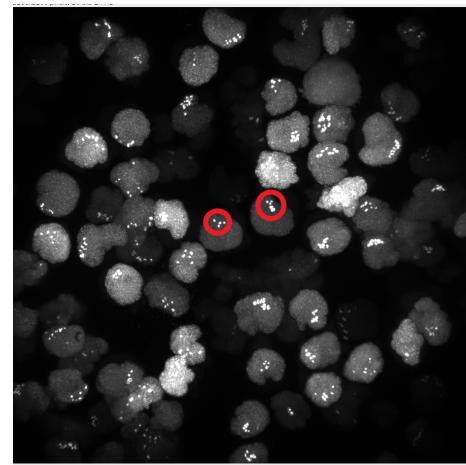
Ce problème se manifestant sur des films de 2 heures, il est probable que sur des films de 12 heures il puisse être amplifié. BigFish basant sa détection des molécules sur des pics d'intensités, une baisse progressive de celle ci va fausser le spot de référence et donc la quantification finale des sites de transcription.

Segmentation et découpage des cellules

Un problème majeur pour l'utilisation de ce pipeline existe dans notre situation. Dans notre cas les cellules se déplacent beaucoup ce qui rend difficile la projection dans le temps et la segmentation.



(a) Projection qui rend difficile le découpage de cellules



(b) Projection qui rend possible le découpage

FIGURE 2.15 – Projection d'un film de 2h en fonction du temps et masques associés

La figure (a) est une projection qui rend difficile le découpage car aucune cellule n'est reconnaissable contrairement à la figure (b)

Il est impossible de reconnaître à l'oeil nu la majorité des cellules lorsqu'on réalise une projection dans le temps (voir Figure 2.15a). Ce qui mène à une mauvaise segmentation avec peu de cellule reconnaissable par Cellpose.

Isolement des cellules

Un autre problème lié au déplacement des cellules, est la présence d'autre cellule à coté de la cellule découpée (voir Figure 2.17). Ce qui peut fausser les résultats de quantification en introduisant des spots et des clusters étrangers qui vont participer aux étapes de détection.



FIGURE 2.17 – Exemple de cellule avec une cellule voisine

Ici le site de transcription de la cellule voisine entourée en rouge est plus intense que celui de la cellule d'intérêt, ce qui fausse la détection.

Limites

Le test de ce pipeline m'a permis de mieux choisir les packages à utiliser et de développer des fonctions pour réaliser des opérations sur les images. (voir Chapitre 3.1.2) De plus nous travaillons sur une problématique relativement nouvelle, la quantification des molécules dans des films de cellule lymphocyte T non adhérentes et vivantes. Certains ressources peuvent suffire pour réaliser des tâches intermédiaires tandis qu'il va falloir mettre en place d'autres méthodes pour s'adapter aux particularités des cellules non adhérentes qui bougent beaucoup pendant l'imagerie.

2.4.2 Choix des ressources

Dans notre cas nous utiliserons Ultrack en combinaison avec Cellpose pour la segmentation.

Pour la quantification nous resterons sur BigFish et BigFishLive.

Pour le plan du pipeline j'ai adapté celui de BigFishLive à notre situation, il comprend donc trois phases distinctes :

- Préparation des films (correction du photobleaching par Image J)
- Segmentation
- Détection et quantification

3 Matériel et Méthodes

3.1 Matériels

3.1.1 Ressources

Pour réaliser ce projet, des langages de programmation, des logiciels et des packages Python sont utilisés.

Les principaux langages de programmation sont :

- Python
- Matlab

Les logiciels utilisés sont :

- ImageJ et Napari pour l'analyse et le traitement d'images
- Matlab R2023 utilisés initialement par l'équipe de MFP
- Jupyter pour l'élaboration de script

Pour le tracking et la quantification des molécules Ultrack et BigFishLive seront essentiels.

Les packages présentés en détail dans le chapitre [2](#) comme :

- Cellpose
- BigFish et BigFishLive
- Ultrack

D'autres packages sont utilisés entre autre :

- Matplotlib pour afficher certaines données
- numpy et dask permettant de réaliser des opérations sur les tableaux

3.1.2 Développement préliminaires

En parallèle de l'essai de BigFishLive, j'ai développé :

- un notebook permettant de faire de la détection pour plusieurs cellules. Ce script correspondant principalement aux scripts originaux qui m'ont été confiés à l'origine mais avec l'utilisation de boucle.
- Un notebook permettant de découper à travers le film les masques de segmentation et les cellules. Ce qui permettra d'appliquer les masques sur les cellules et de cacher les cellules voisines à l'extérieur du masque.
- Un script qui permet de trier les clusters et de ne garder que les clusters correspondant aux régions les plus intenses, il est d'ailleurs possible d'y appliquer un seuil ce qui permet de l'adapter en fonction des films utilisés.

Ce qui fait que je suis capable de :

- Suivre et segmenter les cellules grâce à Ultrack (Etape de correction possible)
- Déetecter et quantifier les spots et clusters grâce à Bigfish
- Corriger les défauts de la détection (clusters surnuméraire)
- Sauvegarder les labels des cellules, les fichiers csv des spots et des clusters.

3.2 Méthodes

3.2.1 Segmentation

Afin de pouvoir obtenir à la fin des films réduits pour chaque cellule, plusieurs étapes sont nécessaires :

- Segmentation avec Cellpose
- Suivi des cellules avec Ultrack
- Correction des masques obtenu avec Ultrack
- Sélection des cellules
- Découpage des cellules

Détection des cellules

Cette étape utilise Cellpose pour segmenter les cellules à travers le film.

Les paramètres utilisés lors de la segmentation sont :

- diameter : Le diamètre des cellules en pixel.

- choix du modèle : 2 modèles sont disponibles "cyto" et "nuclei", afin de segmenter le noyau uniquement ou les cellules entières.

Dans nos cellules les noyaux prennent presque toute la place à l'intérieur des cellules, nous utiliserons donc le modèle "cyto" avec un diamètre de 120.

Tracking

Différent paramètres d'Ultrack existent pour optimiser Ultrack :

- max_distance : Qui correspond à la distance maximale entre les cellules.
- n_workers : Qui correspond au nombre de workers thread.
- appear_weight : Qui correspond au poids de pénalisation pour l'apparition d'une nouvelle cellule.
- disappear_weight : Qui correspond au poids de pénalisation pour la disparition d'une cellule.
- division_weight : Qui correspond au poids de pénalisation pour la division d'une cellule.

Toutes les valeurs liés aux pénalités doivent être négatives.

Puis on a :

- power : Exponentielle η du pouvoir de transformation w_{pq}^η
- solution gap : gap de solution

Les paramètres actuellement choisi sont :

- max_distance = 15
- n_workers = 1
- appear_weight, disappear_weight et division_weight = -0.001
- power = 4
- solution gap = 0.001

Une fois ces étapes réalisées ont obtient des masques sous forme d'images Tif que nous allons sauvegarder.

Correction des masques obtenu avec Ultrack

Parmi toutes les cellules présentes dans le films, pour les étapes suivantes nous allons garder que les cellules vivantes et clairement visible durant tout le film. Ce qui

signifie que nous ne devrons pas utiliser (voir Figure 3.1) :

- Les cellules mortes
- Les cellules coupées car présentes aux bords des images
- Les cellules cachées par d'autres
- Les cellules qui se divisent au cours du film

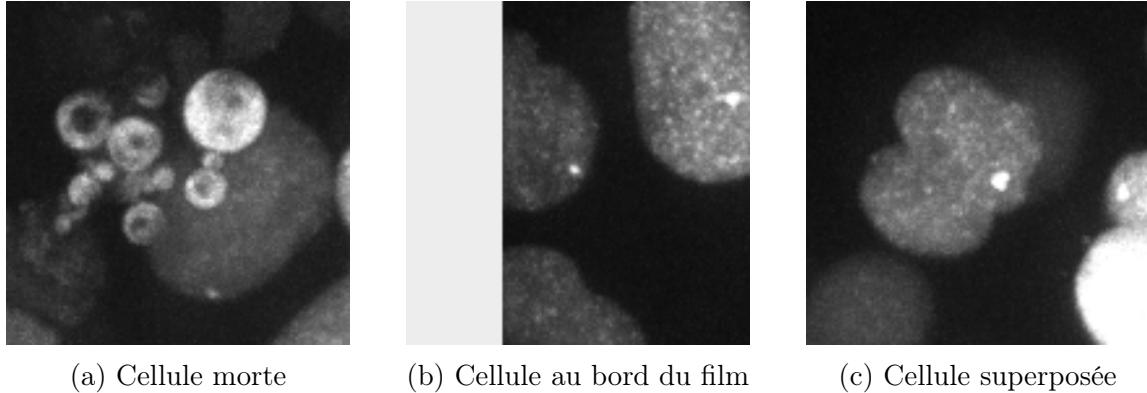


FIGURE 3.1 – Exemple de cellules à éliminer

De plus il est à noter que les étapes de segmentation et de tracking ne sont pas parfaites pour plusieurs raisons :

- Segmentation incomplète d'une cellule, ce qui peut mener à une détection partielle des cellules. (masque tronqué, ou deux masques différents détecte une même cellule)
- Saut de cellule. Lors du tracking le masque de suivi peut changer de cellule d'un instant à un autre ce qui fait que l'on ne suit plus la même cellule.

On peut donc être amenés à corriger manuellement les masques de cellules pour :

- Modifier le suivi des cellules
- Fusionner plusieurs masques correspondant à une même cellule en réalité
- Re-dessiner un masque dont la forme est incorrecte

Pour cela plusieurs fonctions ont été développées afin de d'automatiser le plus possible certaines opérations sur les masques tout le long du film.

Tel que *merge_neighboor_cells* et *delete_cells* qui gèrent respectivement la fusion de masques, et leurs suppression.

Découpage des cellules

Une fois les cellules corrigées une liste correspondant à leurs labels sera sauvegardée sous forme de fichiers npy. Ce qui permettra d'isoler chaque cellule en fonction du temps, de les découper et de les sauvegarder sous forme de stacks. Réduire la dimension des images permet de réduire leur taille et la vitesse d'exécution des scripts.

Pour isoler les images 3D de cellules et les cropper, nous allons utiliser les propriétés des numpy array notamment, les méthodes comme *numpy.where()* et *numpy.roll()*.

Pour cela nous allons procéder en deux étapes :

- Centrage du film sur une cellule d'intérêt en utilisant les masques
- Découpage du film sur la cellule centré

Le centrage sur une cellule est réalisé en calculant le centroïde de la cellule à chaque instant du film.

Une fois cela fait, un centroïde moyen sera calculé et un décalage pour chaque instant du film. Ensuite chaque images sera centré en fonction des décalages calculés.

Une fois une cellule centrée, nous allons découper le film autour de cette cellule afin d'obtenir un film pour une cellule individuelle.

Les fonctions respectives développées pour réaliser ces opérations sont *center_images_for_z* et *crop_cells_with_background*.

3.2.2 Détection et quantification

Une fois les films de cellules découpés, nous pouvons déterminer le nombre de spots et de clusters avec BigFishLive. Pour cela nous allons reprendre les étapes suivantes de BigFishLive :

- Détermination du seuil
- Construction du spot de référence
- Détection de spot et de clusters sur tout le film
- Quantification

Le détail de ces étapes ayant déjà été décrit dans le Chapitre 2 nous allons seulement décrire les améliorations que j'ai développé durant mon stage.

Construction du spot de référence

Comme nous travaillons avec des images de cellules individuelles, il se peut que nous ayons des cellules voisines dans le champ, leurs présence risque d'affecter la qualité du spot de référence et donc la détection et la quantification des spots et des clusters (voir Figure 3.2). Pour éviter cela nous allons utiliser les propriétés des masques. Pour cela nous allons découper en plus des cellules les masques associés afin de les ré-utiliser pour ne conserver que les cellules présentes à l'intérieur en modifiant la valeur des pixels à l'extérieur de ce masque.



FIGURE 3.2 – Cellule 24 du film c3_21_c3_45_ON_1_w1SPI488_s3 avec et sans cellule voisine

Nous allons faire ça à deux étapes clés :

- Avant la détection des spots préliminaire
- Avant la détection des spots et des clusters

Il a aussi fallu mettre en place des solutions dans le cas de cellule où il y a très peu de transcription et donc très peu de molécules à l'intérieur.

Dans ce genre de cellule, on peut retrouver des faux spots présent au bord de la cellule. Leur présence peut pour effet de fausser le spot de référence et donc la quantification des sites.

Détection de spot et de clusters

Une fois la détection faite on obtient une liste de spots et de clusters. Le site de transcription est généralement le point le plus intense de la cellule. Pour raccourcir le pipeline original, j'ai développé *order_clusters_frames*, une fonction permettant de trier la liste des clusters en fonction de son intensité. Cette fonction prend en compte un seuil d'intensité, cela signifie que si aucun cluster ne dépasse ce seuil, on considère le site de transcription comme absent de la cellule. La fonction renverra une liste des clusters avec des lignes remplis de 0 aux endroits correspondant.

De plus, il est parfois possible que le point le plus intense de notre image ne corresponde pas au site, il est possible de choisir manuellement un autre cluster.

Quantification

Les sites de transcriptions étant une structure tridimensionnelle et de forme changeante, il se peut qu'il y ait de multiples détections de clusters par BigFish à des endroits très rapprochés dans le cas de gros sites.

Pour cela, la fonction *merge_clusters_frame* a été développée et prend en paramètres :

- une liste de clusters
- La projection 2D du film
- La distance en xy et la distance en z, en pixel
- Une marge de tolérance basé sur l'intensité des clusters. Cela permet d'éviter d'ajouter un cluster n'appartenant pas au site de transcription.

Par exemple pour :

- un pourcentage de 0.5
- une distance xy = 5
- une distance z = 1

Ne seront additionnés que les clusters se trouvant à une distance inférieure au point le plus intense et dont l'intensité minimale correspond à la moitié. De plus pour éviter de prendre en compte des candidats n'appartenant pas au site de transcription, une

marge de tolérance sera calculée à partir du pixel le plus intense.

4 Résultats et Discussion

4.1 Résultats

Une fois la détection réalisée on obtient des fichiers csv contenant les informations correspondant aux clusters, allant des coordonnées aux nombre de molécules.

z	y	x	n_molecule	minute
0.0	0.0	0.0	0.0	0.0
14.0	64.0	104.0	3.0	1.0
17.0	39.0	100.0	3.0	2.0
34.0	54.0	93.0	6.0	3.0
17.0	40.0	116.0	6.0	4.0
20.0	43.0	150.0	6.0	5.0
18.0	49.0	142.0	6.0	6.0
8.0	45.0	116.0	4.0	7.0
20.0	34.0	73.0	3.0	8.0
14.0	26.0	93.0	5.0	9.0
32.0	23.0	119.0	6.0	10.0

FIGURE 4.1 – Extrait du fichier csv de la cellule 56 du film C3.14_ON_01_w1SPI-488_s4

Ce résultat a été obtenu avec alpha = 0.5, beta = 2, gamma = 15

Pour vérifier nos résultats nous allons principalement regarder les films et vérifier :

- les coordonnées des sites de transcription
- Le profil des sites sur les films. Une augmentation du nombre de molécule au site de transcription peut être assimilé à un élargissement de ce site et vice versa.

Mais cette méthode reste imparfaite car nous utilisons une projection en 2D pour vérifier la quantification d'un objet 3D.

4.2 Discussions

Pour contrôler nos résultats différentes manières sont possibles :

- Changer un ou plusieurs paramètre au cours de la détection

4.2.1 Changement de paramètres

Une fois le seuil déterminé, les paramètres sur lesquels nous pouvons agir sont :

- alpha (de la fonction *buildReferenceSpotFromImages* de BigFishLive)
- beta et gamma (de la fonction *getSpotAndClusters* de BigFishLive)
- La distance xy et z (de la fonction *merge_clusters_frame* que j'ai développé)
- Le pourcentage de tolérance (aussi de *merge_clusters_frame*)

alpha :

Nous avons testé 3 valeurs d'alpha : 0.3, 0.5 et 0.9. Plus alpha sera élevé plus le nombre de molécules au niveau des sites de transcriptions sera faible. Par exemple entre un alpha de 0.3 et 0.5 nous pouvons avoir une différence du simple au double.

Récemment, Rachel Topno nous a envoyé un notebook permettant de choisir un alpha pour une cellule. Le notebook va réaliser une détection préalable de molécules uniques d'ARN, puis il va calculer un score qualité des spots par rapport au spot de référence. Les vrais spots auront un score proche de 1, tandis que les faux positifs auront une valeur très supérieure ou très inférieure à 1.

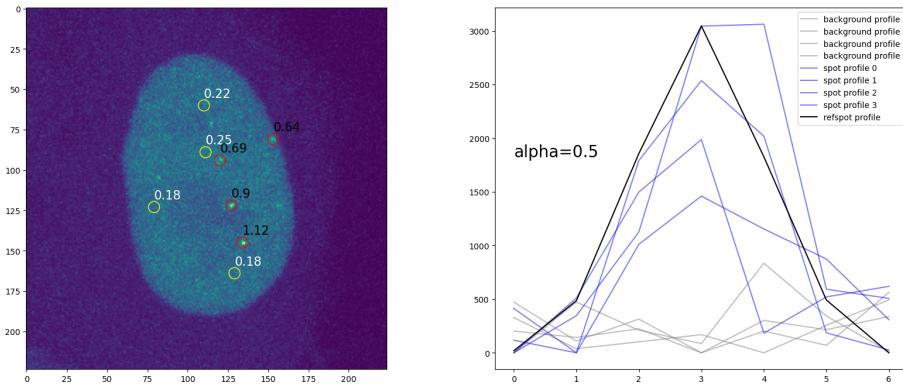


FIGURE 4.2 – Score de qualité des spots pour une image et profil des spots correspondant (données de démonstration de BigFishLive)

Les faux positifs sont entourés en jaune, et les vrais spots sont entourés en rouge.

Ensuite, le notebook donné va tester différentes valeurs de alpha et calculer un score moyen de qualité pour chaque alpha en fonction du temps.

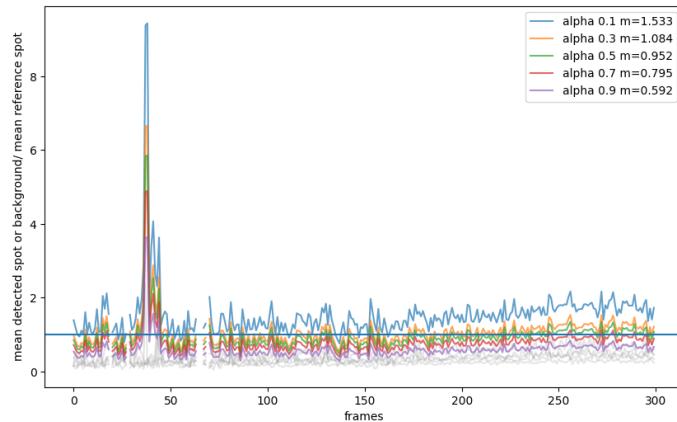


FIGURE 4.3 – Rapport de la détection moyenne des spots / spot de référence en fonction du temps (données de démonstration de BigFishLive)

Ici la valeur d'alpha optimale est alpha=0.5

beta et gamma :

Quant à beta, passer de 1 à 2, n'impacte pas significativement les résultats de quantification. Dans le cas de gamma, nous avons testé 3 valeurs, 5, 10 et 15. Nous

estimons que les valeurs de gamma supérieurs à 10 nous donnaient les résultats les plus pertinent.

Conclusion

A la fin j'ai donc un pipeline permettant de :

- Segmenter et suivre des cellules
- Les découper et les centrer dans des films
- Et capable de réaliser la détection de molécules uniques et la quantification du nombre de molécules aux sites de transcription.

4.2.2 Application et Perspectives

Application

Ce pipeline peut avoir d'autres applications, il peut être utilisé sur d'autres type cellulaires et pour quantifier n'importe quel spots contenant plusieurs fluorophores à l'intérieur de cellules.

Perspectives

Plusieurs perspectives sont en cours pour ce pipeline :

- Recouper la quantification des clusters par d'autres méthodes
- Rendre le pipeline plus user-friendly
- Utiliser ce pipeline sur des films de 12 heures

Pour vérifier si la quantification est fiable, on pourrait par exemple utiliser FISHQuant afin de comparer les résultats pour une cellule à des moments clés. Même si il est à noter que connaître le nombre exact des molécules présent au sites de transcription n'est pas le plus important. L'important est de calculer les variations des nombres de molécules au cours du temps. Et dans ce cas la sur la plupart des cellules où le nombre de molécules détectés est suffisant, le profil des sites de transcription est juste. Les variations des nombres de molécules suivent les films étudiés.

Une autre perspective pour le pipeline serait de transformer le pipeline en un plugin pouvant être entièrement utilisé dans Napari. Cela permettrait de rendre

son utilisation plus user-friendly tout en étant plus simple que de créer un nouveau logiciel. Les principales fonction utilisée dans BigFish, BigFishLive et celles que j'ai développé étant dans des fichiers .py il sera donc possible de les importer et des les utiliser dans Napari pour réaliser les opération relatifs au pipeline.

5 Remerciement

Un grand merci à toute l'équipe Andevir Team du MFP : HIV Transcription Project :

- Eugenia Basyuk
- Alexia Damour
- Marie-Line Andreola
- Mathieu Metifiot
- Patricia Recordon-Pinson
- Floriane Lagadec
- Chloe Torres
- Akintha Ganot
- Enola Reaud

A Nicolas Landrein du MFP SpacVir team

A nos collaborateurs :

- Rachel Topno
- Edouard Bertrand

Et a Marie-Beurton Aimar du LABRI.

Le temps de calcul pour cette étude a été fourni par les installations informatiques du MCIA (Mésocentre de Calcul Intensif Aquitain).

Bibliographie

- [1] 'Cell Segmentation'. URL : <https://paperswithcode.com/task/cell-segmentation>.
- [2] 'VIH et sida'. URL : '<https://www.who.int/fr/news-room/fact-sheets/detail/hiv-aids#:~:text=On%20estimait%20%C3%A0%2039%C2%2C0,R%C3%A9gion%20africaine%20de%20l'OMS.>'.
- [3] Analysis of smFISH images! FISH-quant. URL : <https://fish-quant.github.io/>. (accessed : 11.01.2024).
- [4] Jordão BRAGANTINI, Merlin LANGE et Loïc ROYER. *Large-Scale Multi-Hypotheses Cell Tracking Using Ultrametric Contours Maps*. 2023. arXiv : [2308.04526 \[cs.CV\]](https://arxiv.org/abs/2308.04526).
- [5] Takamasa Kudo DAVID A. VAN VALEN. "Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments". In : *PLOS computational biology* (2016). DOI : <https://doi.org/10.1371/journal.pcbi.1005177>.
- [6] Emanuele FANALES-BELASIO et Mariangela RAIMONDO. "HIV virology and pathogenetic mechanisms of infection : a brief overview". In : *Annali dell'Istituto Superiore di Sanità* (2010). DOI : https://doi.org/10.4415/ann_10_01_02.
- [7] napari : a fast, interactive viewer for multi-dimensional images in Python. URL : <https://napari.org/stable/>. (accessed : 15.01.2024).
- [8] Uwe SCHMIDT et al. "Cell Detection with Star-Convex Polygons". In : (2018), p. 265-273. DOI : [10.1007/978-3-030-00934-2_30](https://doi.org/10.1007/978-3-030-00934-2_30).
- [9] Fabienne Rayne SOLÈNE DEBAISIEUX. "The Ins and Outs of HIV-1 Tat". In : *John Wiley Sons A/S* (2011). DOI : <https://doi.org/10.1111/j.1600-0854.2011.01286.x>.

- [10] STRINGER et CARSEN. “Cellpose : a generalist algorithm for cellular segmentation”. In : *Nature Methods* (2021). DOI : <https://doi.org/10.1038/s41592-020-01018-x>.

6 Annexe

En plus de travailler sur des films de cellule vivantes, j'ai aussi été amené à travailler sur des cellules fixes. Mon rôle consistait principalement à réaliser de la segmentation afin d'obtenir les contours (outlines) de cellules sur MATLAB.

Les images de cellules fixes diffèrent de celles des cellules vivantes.

6.1 Jeu de données

Dans le cas des cellules fixes, l'équipe utilise la technique smFISH pour marquer les molécules d'ARN transcrites (single molecule Fluorescence In Situ Hybridization). smFISH utilise une sonde d'oligonucléotide couplé avec CY5 (sonde ADN simple brin) qui va s'hybrider avec l'ARN et le rendre visible en fluorescence.

Un microscope de type "spinning disk" est utilisé.

Deux principaux paramètres sont utilisés pour acquérir des images :

- La puissance des lasers, généralement à 60%
- Le temps d'acquisition généralement à 200ms

L'équipe utilise les longueurs d'ondes suivantes et les marqueurs associées :

- 642 nm : FISH CY5 (rouge lointain) qui détecte aussi l'ARN du VIH
- 562 nm : Scarlet-Tat(rouge) qui marque les noyaux de cellules
- 488 nm : MCP-SG (vert) pour suivre les molécules d'ARNm dans les cellules vivantes
- 405 nm : DAPI (bleu) qui permet de marquer les noyau de cellules fixes

4 conditions sont utilisées pour tester le niveau de transcription des clones :

- NA
- PMA
- TNF
- SAHA

Tous les smFISH doivent être fait avec les mêmes paramètres, sauf les images prises avec le DAPI. En raison de leurs qualités inconstantes, on peut être amené à jouer sur ces paramètres afin d'obtenir de meilleures images.

Les critères de sélection des images sont les suivants :

- Il faut avoir le maximum de cellules visibles pour pouvoir les analyser.
- Et il faut avoir le marquage le plus qualitatif possible.

Pour un clone dans une condition donnée on obtient 16 images correspondant aux :

- 4 types d'images dépendant de la longueur d'onde des molécules utilisées.
- et pour chaque longueur d'onde on prendra des captures dans 4 localisations différentes de préférence là où il y a beaucoup de cellules vivantes afin de pouvoir obtenir l'échantillon le plus large possible afin d'obtenir les modèles statistiques les plus pertinents possibles.

6.2 Obtention des outlines

6.2.1 Pipeline

Le pipeline consiste à :

- Projeter les images en 2D
- Segmenter les images
- Créer les outlines sur FISHQuant à partir des masques de segmentation
- Modifier les outlines afin de les rendre lisibles par FISHQuant

Projection des images

Cette fois-ci la projection des images est réalisée sur ImageJ, avec un script qui permettra de projeter les images en 2D et de les sauvegarder dans un dossier à part.

Segmentation des images

La segmentation sera réalisé par Cellpose qui va nous donner les masques de segmentation sous cette forme.

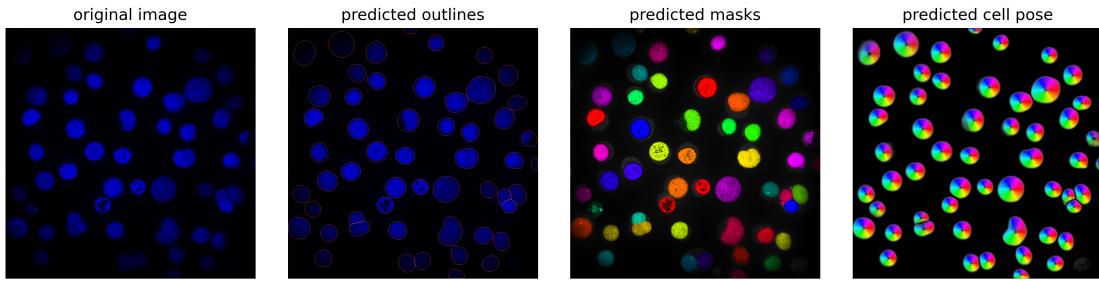


FIGURE 6.1 – Masque de segmentation du clone3.14_JQ1_01_w3SPI-488

Création des outlines

La création des outlines se fera sur FISHQuant, pour cela nous allons utiliser le CellProfiler. Sur le CellProfiler il faut dabord introduire les paramètres suivants :

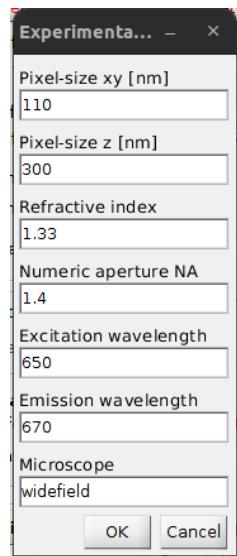


FIGURE 6.2 – paramètre du CellProfiler

Une fois les paramètres introduits nous chargeons les images et les masques de cette façon :

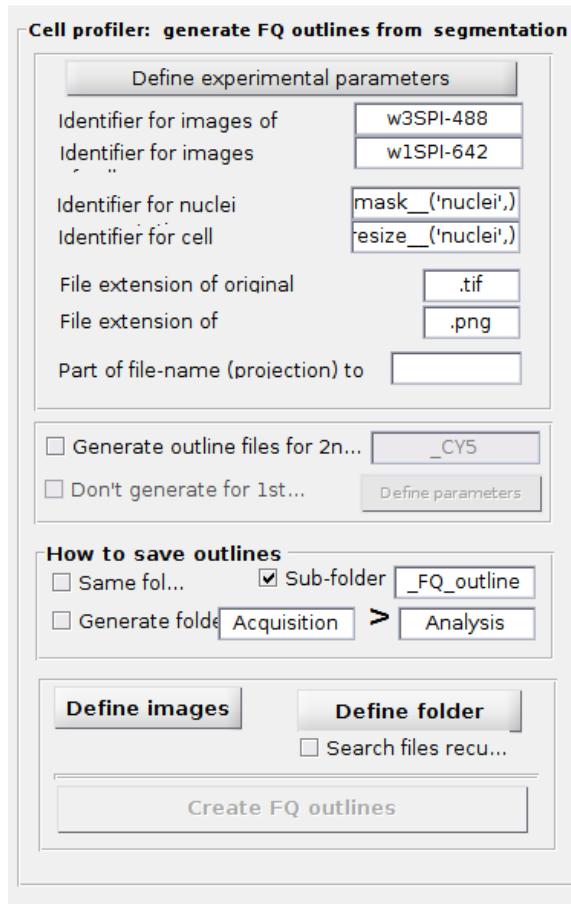


FIGURE 6.3 – Cell Profiler

Pour la segmentation nous allons généralement utiliser les images obtenues avec le w3SPI-488 ou le cas échéant celles obtenues avec le w4SPI-405. Une fois cela fait cela va permettre de générer les outlines.

Modification des outlines

Les outlines ont la forme de fichier texte avec des informations sur les films, les cellules et les noyaux détectés.

```

1 FISH-QUANT      v3
2 File-version    3D_v1
3 RESULTS OF SPOT DETECTION PERFORMED ON 12-Apr-2024
4 COMMENT Automated outline definition (batch or quick-save)
5 IMG_Raw clone3.24_NA_05_w1SPI-642.tif
6 IMG_Filtered
7 IMG_DAPI        clone3.24_NA_05_w3SPI-488.tif
8 IMG_TS_label
9 FILE_settings
10 PARAMETERS
11 Pix-XY  Pix-Z  RI   Ex   Em   NA   Type
12 110     300    1.33  650   670   1.4   widefield
13 CELL_START Cell_CP_1
14 X_POS   396   397   398   399   400   401   402   403   404   405   405   406
407     408   409   410   411   412   413   414   415   416   417   418   419
420     421   422   423   424   425   426   427   428   429   430   431   432
433     434   435   436   437   438   439   440   441   442   443   444
445     446   447   448   449   450   451   452   453   454   455   456   457
458     459   460   461   462   463   464   465   466   467   468   469   465
465     465   465   464   464   463   463   462   462   461   460   459   458
457     456   455   454   453   452   451   450   449   448   447   446   445
444     443   442   441   440   439   438   437   436   435   434   433   432
431     430   429   428   427   426   425   424   423   422   421   420   419
418     417   416   415   414   413   412   411   410   409   408   407   406
405     405   404   403   402   402   401   400   399   399   399   399   399
398     398   397   397   397   396   396   396   396   396   396   396   396
15 Y_POS   6     5     5     4     4     4     3     3     3     2     1     1
1       1     1     1     1     1     1     1     1     1     1     1     1
1       1     1     1     1     1     1     1     1     1     1     1     1
1       1     1     1     1     1     2     3     3     3     3     3     3
3       3     3     3     3     3     3     3     3     3     3     3     3
3       3     4     4     4     4     5     5     6     7     8     9     10
11      12    13    14    15    16    17    18    19    20    21    21
22      22    23    24    25    26    27    27    28    28    29    29    29
30      30    31    31    32    32    33    34    34    35    35    35    35

```

FIGURE 6.4 – Exemple d’outlines pour le clone3.14_JQ1_01_w3SPI-488

Dans notre cas pour permettre la lecture de ces fichiers par FISHQuant il faut modifier :

- Le titre du fichier
- Et la ligne 5 du fichier qui correspond au nom du fichier pour l’image d’origine.

Nous allons simplement modifier la longueur d’onde utilisée pour ces images par celle de w1SPI-642.

Analyse des images par FISHQuant

Une fois cela fait ont pourra simplement charger les dossiers contenant les images et les outlines dans FISHQuant pour commencer l’analyse des images.

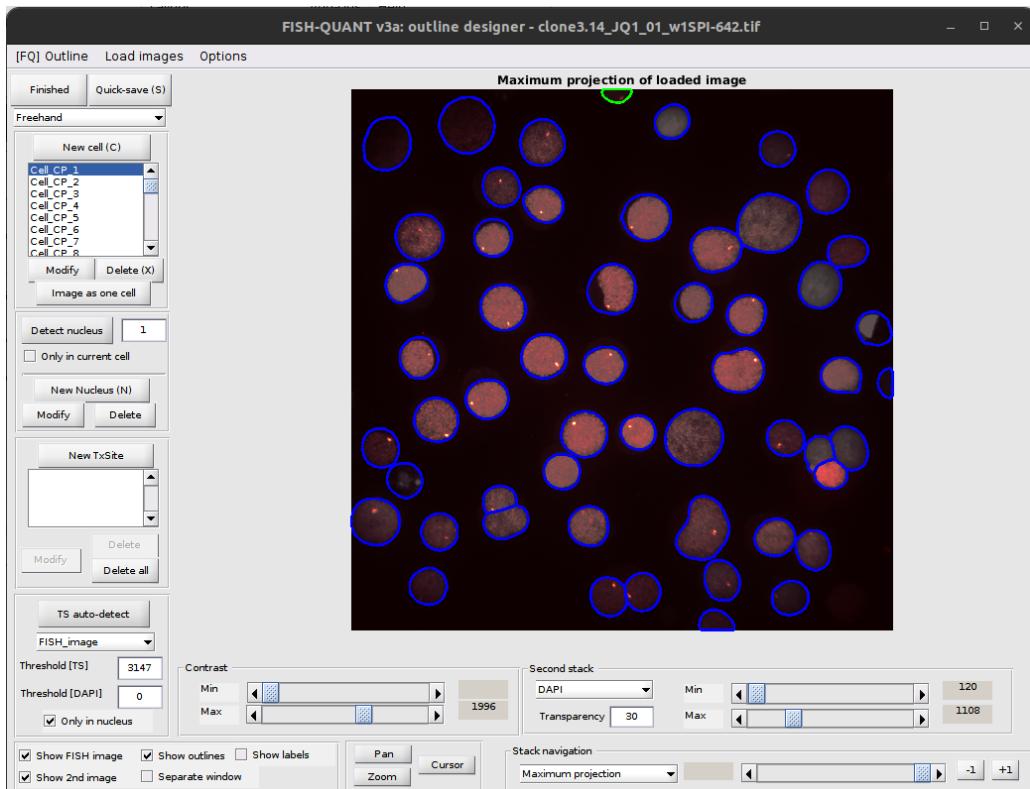


FIGURE 6.5 – Outline Designer de FISHQuant avec l'image du clone3.14_JQ1_01_w3SPI-488

L'image FISH est superposée avec celle obtenue en DAPI et le contour des cellules