

# Rapport - Traitement d'images numériques

TANG Elody

Novembre 2022

# 1 Introduction à l'application ImageJ

L'objectif de ce TP est de se familiariser avec l'application ImageJ, qui est un logiciel de traitement d'image. À l'aide de ses outils, nous pouvons produire plusieurs effets différents sur des images comme la convolution numérique. ImageJ est un logiciel du domaine public et au code source ouvert. Il est écrit en Java.

Il y a la possibilité de programmer une chaîne de traitements en écrivant des macros et étendre les capacités du logiciel avec de nombreux plugins. Le logiciel travaille sur des images de huit bits et n'affiche que des résultats entiers positifs. Le niveau de gris noir est représenté par la valeur **0**, et le niveau de gris blanc est représenté par la valeur **255**.

## 1.1 Lancement du logiciel

Il y a deux possibilités de lancer ce logiciel :

- Par la commande sur le terminal : **imagej**
- Par le mode graphique : **Menu/Graphiques/ImageJ**

## 1.2 Définitions utiles

**Région d'intérêt (ROI) :** Une région d'intérêt est un sous-ensemble d'une image identifié dans un but particulier. Le logiciel permet de choisir, analyser et sauvegarder des régions d'intérêt dans une image.

## 1.3 Exemple d'application

### 1.3.1 Comptage d'objet

Afin de tester les différents outils de ImageJ, nous allons compter les différents objets de l'image ci-dessous, qui représentent des lentilles.

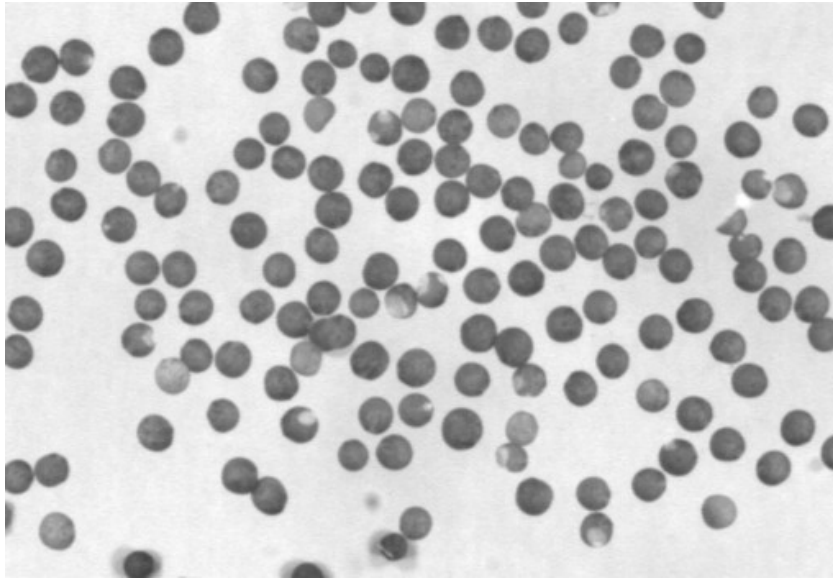


Image originale de lentilles

Pour avoir le nombre total de lentilles présentes sur l'image, nous allons procéder de cette façon :

**1** - Utilisons l'outil **Tracing Tool** (Menu Process / Binary / make Binary) pour avoir une image seuillée. Tous les pixels d'un niveau de gris inférieur au niveau de gris seuil sera d'une valeur **0**, alors que tous les pixels d'un niveau de gris supérieur au niveau de gris seuil sera d'une valeur de **255** (blanc). En d'autres mots, cet outil convertit une image en noir et blanc.

D'après les paramètres du logiciel, le seuil est déterminé à partir de la fonction suivante :

$$Seuil = \frac{MF + MO}{2}$$

avec  $MF$  = Niveau de gris moyen du fond de l'image et  $MO$  = Niveau de gris moyen des objets de l'image.

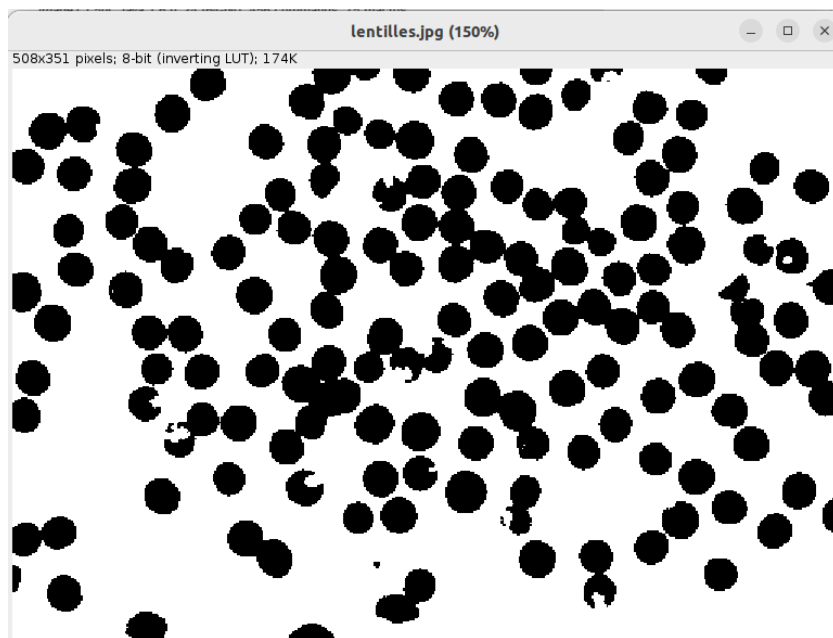
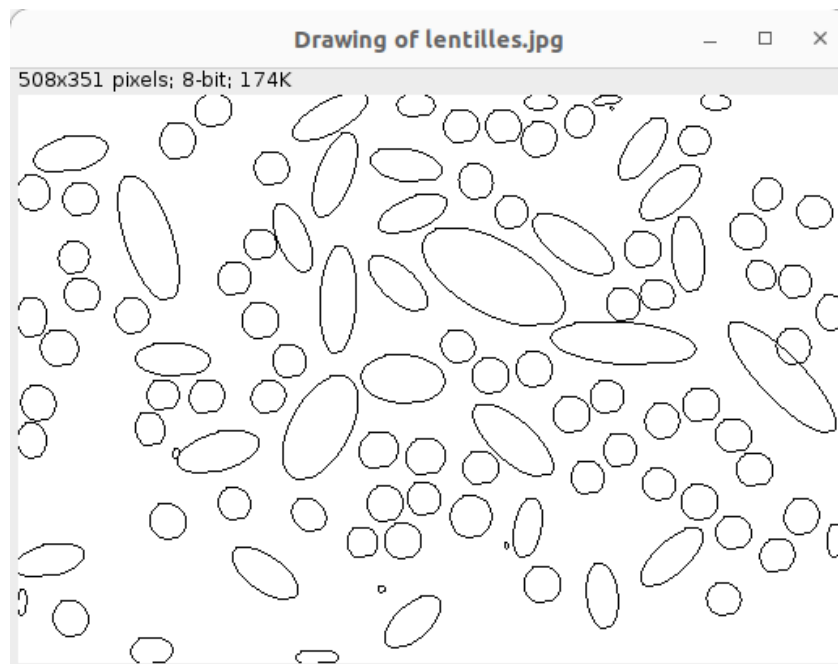


Image seuillée des lentilles

**2** - On effectue la mesure sur l'image entière avec l'outil **Analyze Particles** (Analyze / Analyze Particles). On sélectionne **Ellipses** dans le champ **Show** et on affiche les résultats :



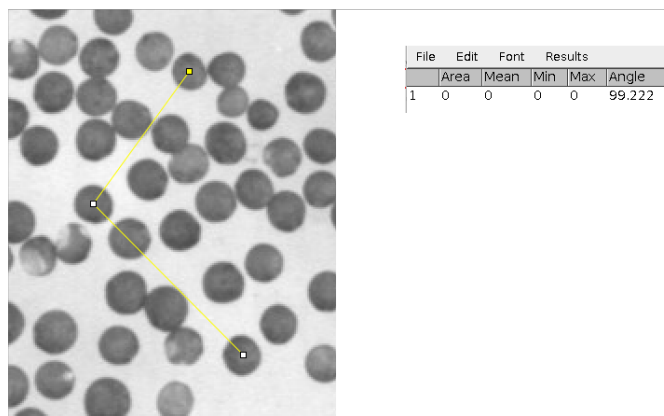
Contour des lentilles

Sur cette image, on peut bien distinguer les lentilles. Cependant, nous ne pouvons pas bien les compter : les lentilles ne sont pas bien réparties et certaines se superposent. Le logiciel va les confondre et les fusionner en un seul objet, comme sur l'image ci-dessus. Il en comptabilise **106**, alors qu'en réalité, il y en a beaucoup plus.

### 1.3.2 Mesures et angles

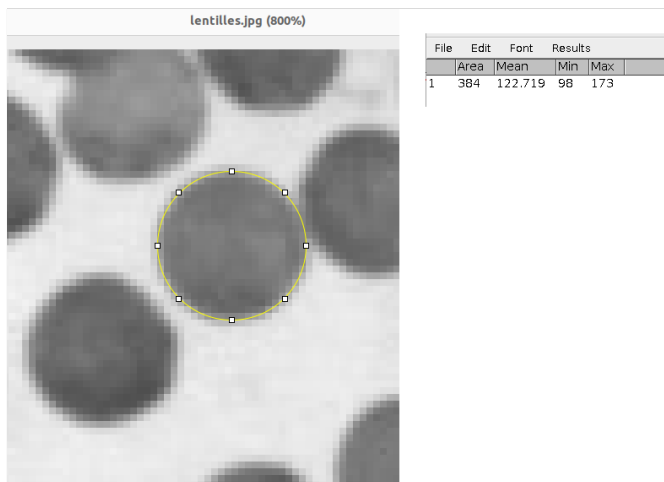
Continuons avec nos tests pour nous familiariser avec le logiciel. Sur la même image d'origine des lentilles, nous allons déterminer l'angle entre trois lentilles et la surface de chacune des lentilles.

Pour déterminer l'angle, utilisons l'outil **Angle Tool** (visible sur la barre d'état) et choisissons trois lentilles au hasard. Un exemple sera affiché un peu plus bas. Ensuite, utilisons l'outil **Measure** (Analyze / Measure). Nous avons ainsi les résultats : l'angle pour ces trois lentilles est de **99.22°**.



Sélection des trois lentilles et son angle

Pour déterminer la surface d'une lentille (qui est aussi l'objet principal de cette image), sélectionnons l'une des lentilles avec l'outil **Oval, elliptical selection** (visible sur la barre d'état), et de la même manière, utilisons l'outil **Measure** (Analyze / Measure). Les résultats apparaissent : pour cette lentille particulière, sa surface est de **384 pixels**.



Sélection d'une lentille particulière et sa surface

Le logiciel est très puissant : il peut délimiter les régions d'intérêt d'une image (et calculer le niveau de gris seuil pour délimiter le fond et les objets) et faire différents types de mesures sur celle-ci. Mais le logiciel est aussi très limité : comme l'image est en 2D, il est incapable de discerner les contours parfaitement d'un objet. Comme on l'a vu sur l'image précédente, ImageJ n'a pas pu compter le bon nombre de lentilles !

## 2 Histogramme

Maintenant, nous allons analyser les histogrammes des niveaux de gris de deux images et voir ce qu'ils apportent comme informations sur l'apparence de leurs images.

### 2.1 Comparaison entre deux images originales

Ici, nous allons tracer les histogrammes de deux images en utilisant l'outil **Histogram** (Analyze / Histogram) de ImageJ. Reprenons l'image des lentilles comme première image et utilisons une image de spermatozoïdes comme seconde image (image d'origine ci-dessous).

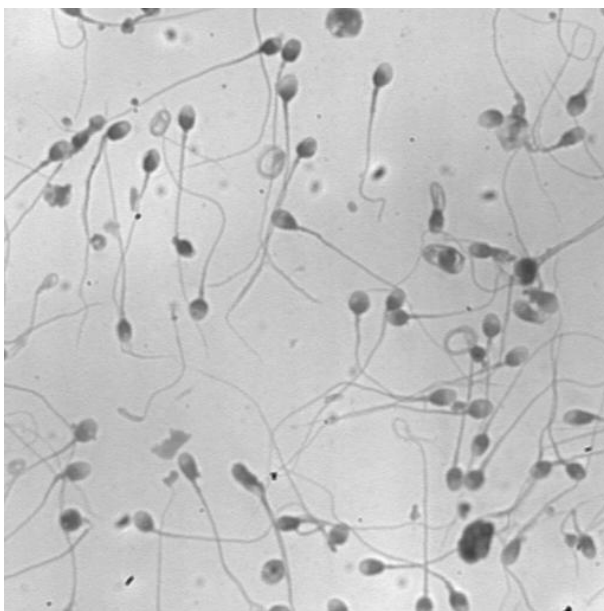
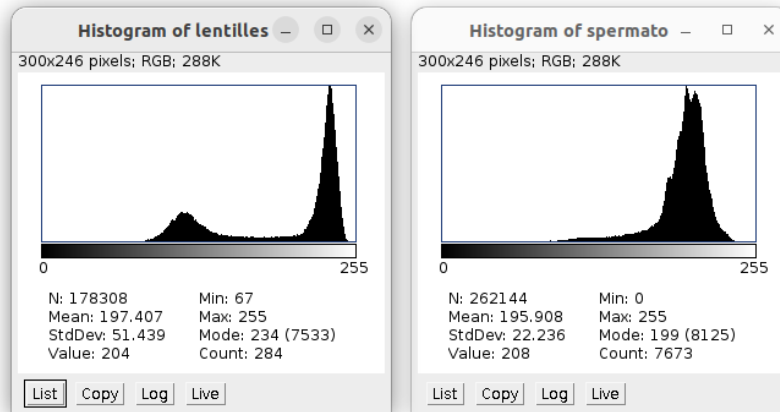


Image originale des spermatozoïdes

Traçons les histogrammes.



- (1) Histogramme de l'image des lentilles
- (2) Histogramme de l'image des spermatozoïdes

Les deux histogrammes nous donnent beaucoup d'informations comme le nombre de pixels total, la moyenne du niveau de gris, la valeur minimale et maximale du niveau de gris...

Sur l'histogramme des lentilles, il y a un petit pic du nombre de pixels vers la valeur **110** : quand on regarde l'image, on peut remarquer le contraste entre les lentilles et le fond de l'image est correcte. Les objets sont facilement reconnaissables, mais le contraste n'est pas optimal, ce qui peut nous permettre de faire un seuillage (1.3.2) La luminosité quant à elle, est légèrement élevée.

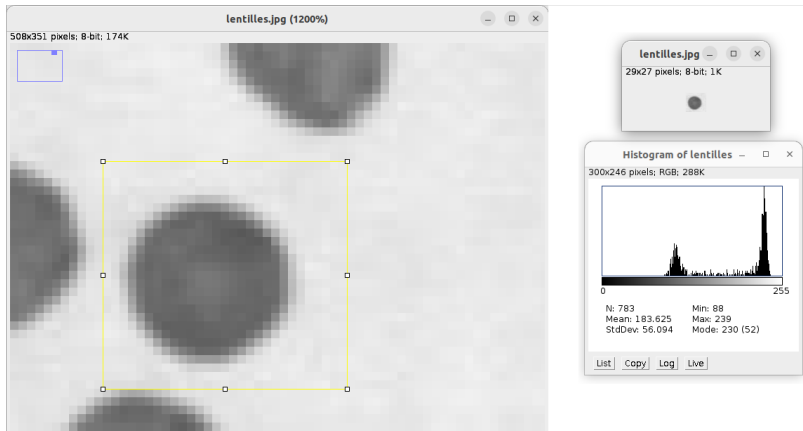
Sur l'histogramme des spermatozoïdes, le contraste est faible tandis que la luminosité est aussi légèrement élevée : nous pouvons reconnaître les objets, mais nous pouvons aussi en déduire que l'image est très claire, comparée à l'image des lentilles.

Maintenant, nous allons analyser le lien entre l'histogramme d'une image originale et l'histogramme d'une ROI dans cette même image. L'intérêt est de voir la différence entre les deux histogrammes.



## 2.2 Comparaison entre une RIO et une image entière

Pour obtenir un histogramme d'une RIO dans une image, il faut tout d'abord délimiter une ROI rectangulaire (pour avoir l'objet et le fond) avec l'outil suivant **Rectangle, Rounded rect selection**. Une fois délimité, il faut rogner l'image pour avoir une deuxième image contenant uniquement la RIO. Pour l'enregistrer, il suffit d'utiliser l'outil **Crop** (Image / Crop) et puis, de la même manière, utiliser l'outil **Histogramm** (Analyze / Histogram) pour avoir son histogramme de niveaux de gris.



Sélection d'une RIO et histogramme

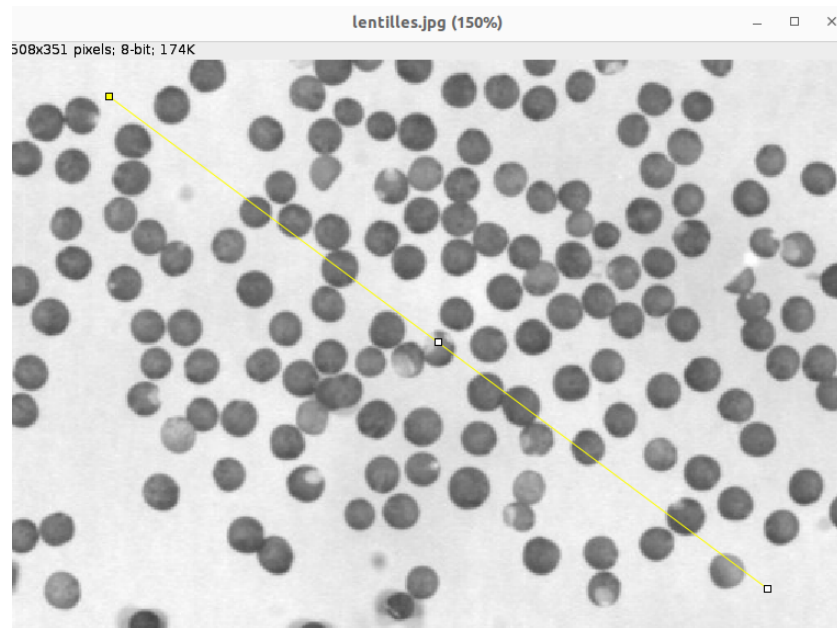
L'historgramme de la RIO de l'image des lentilles est très similaire à l'historgramme de l'image entière. Naturellement, il y a moins de pixels, donc le nombre de pixels est plus petit. Cependant, pour cette RIO, le contraste est plus marquée.

Pour distinguer une RIO du fond d'une image, nous voulons observer la différence de couleur entre les RIO et le fond d'une image. Pour cette raison, utilisons les profils en 2D et en 3D.

## 3 Profils en 2D et en 3D

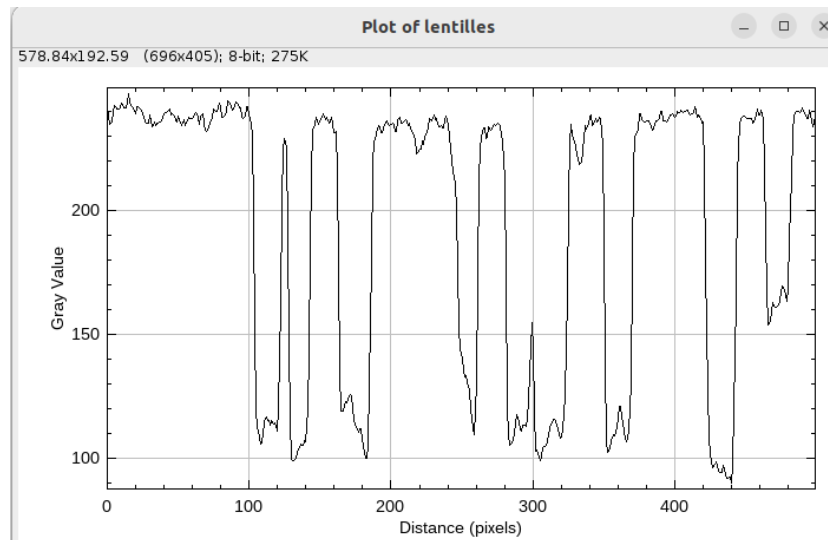
### 3.1 Profil d'une ligne en 2D

En première partie, regardons le profil 2D d'une ligne tracée sur l'image des lentilles. Utilisons l'outil **Straight Line** (visible sur la barre d'état) pour tracer une ligne sur l'image.



Sélection avec un trait sur l'image des lentilles

Pour avoir le profil exact de l'image, nous avons choisi de tracer notre trait en diagonale. Ainsi, notre trait est composé de plusieurs lentilles ainsi que du fond. Ensuite, utilisons l'outil **Plot Profile** (Analyze / Plot Profile) pour afficher le profil d'une ligne. Le résultat (ci-dessous) est un graphe, avec le niveau de gris en ordonnée et la distance entre les pixels en abscisse.



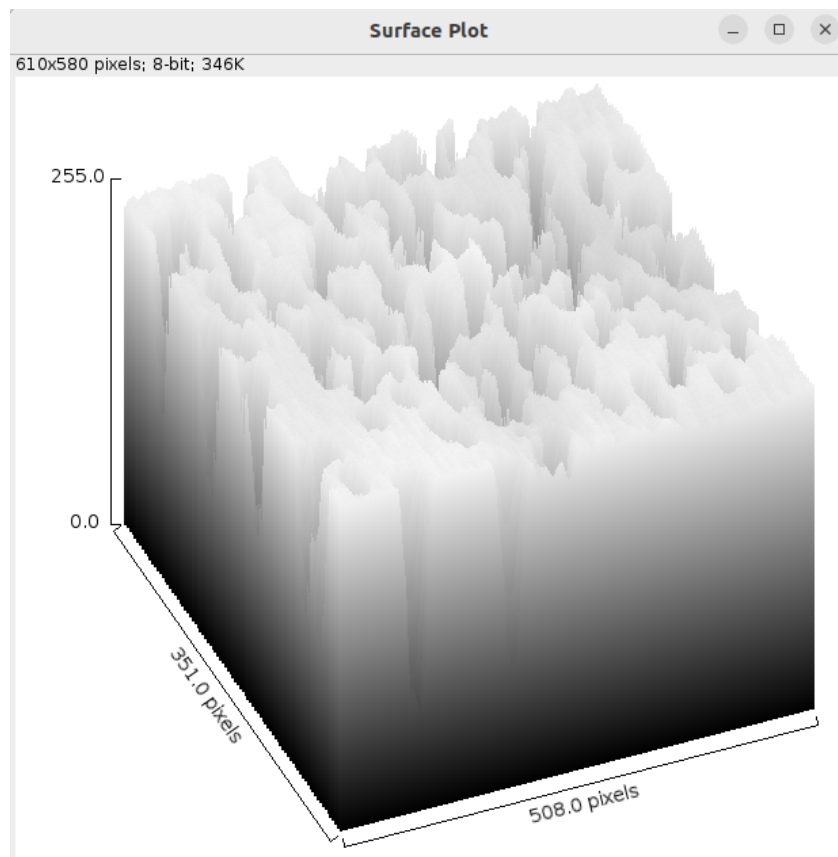
Profil de la ligne tracée

On peut remarquer que le niveau de gris du fond est d'environ **240**, ce qui correspond à un fond plutôt clair, et que la couleur des lentilles est plus foncée, ce qui crée des "trous" dans la courbe. Ici, on peut compter le nombre de lentilles traversant le trait de sélection, ce qui correspond exactement au nombre de "trous". Il y en a exactement **9**.

Maintenant qu'on peut distinguer les lentilles sur un profil d'une ligne, voyons si nous pouvons faire la même chose sur le profil d'une surface en 3D.

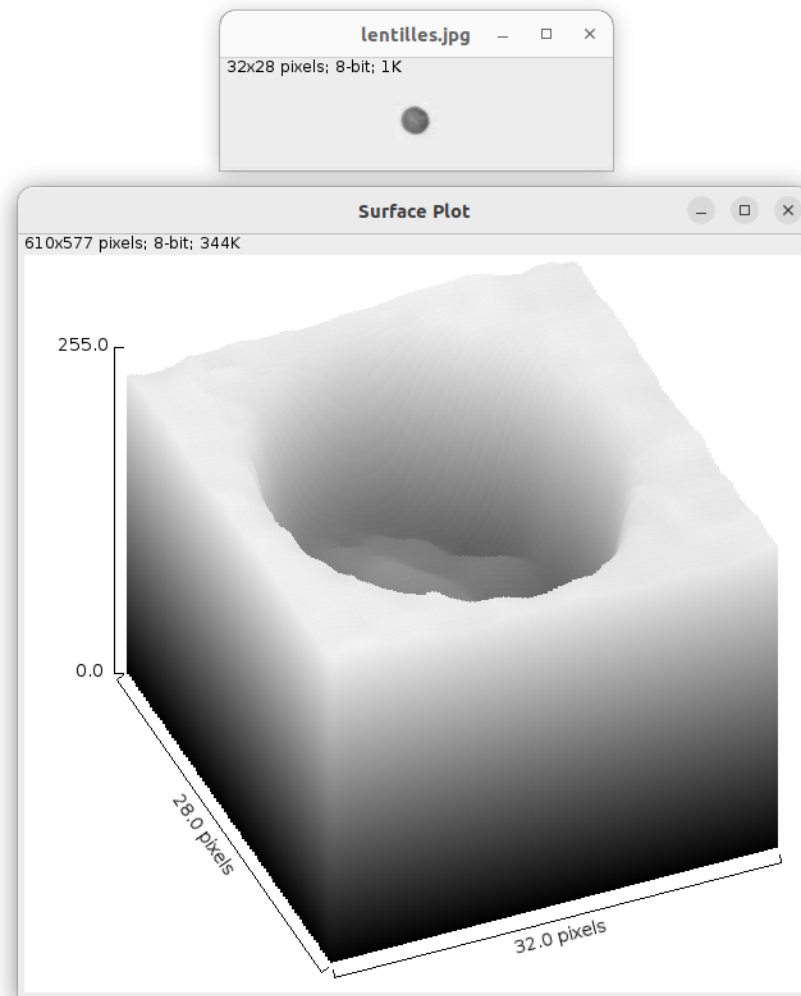
### 3.2 Profil d'une surface en 3D

Pour cela, utilisons l'image d'origine des lentilles et appliquons l'outil **Surface Plot** (Analyze / Surface Plot).



Profil de la surface en 3D de l'image des lentilles

De la même manière, utilisons le même outil sur une RIO de l'image des lentilles. Comme vu précédemment, pour sélectionner une RIO, utilisons les outils suivants : **Rectangle**, **Rounded Rect** et **Crop**.



- (1) Image rognée d'une RIO de l'image des lentilles
- (2) Profil de la surface en 3D d'une RIO d'une image des lentilles

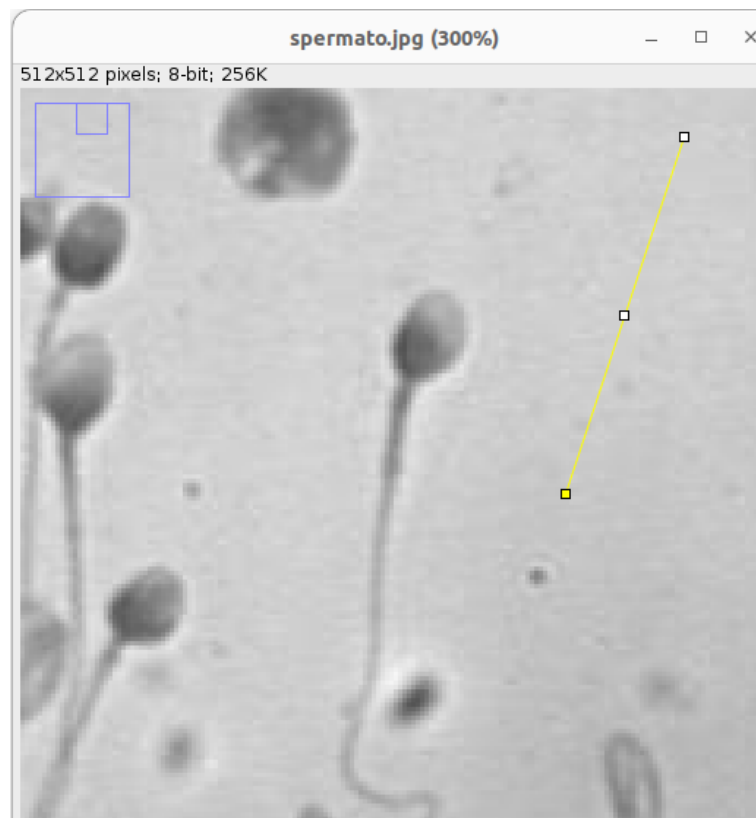
Comme pour le profil d'une ligne en 2D, nous pouvons constater qu'une lentille est caractérisée par un grand "trou". Et sur le profil de la surface de l'image des lentilles, il y a beaucoup de grand "trous", d'où la présence de beaucoup de lentilles. Certains "trous" fusionnent et forment une crevace : ce qui est normal puisque certaines lentilles sont superposées.

Pour distinguer les objets au premier plan et le fond à l'arrière plan, ImageJ se sert des particularités de l'image, notamment la couleur du fond

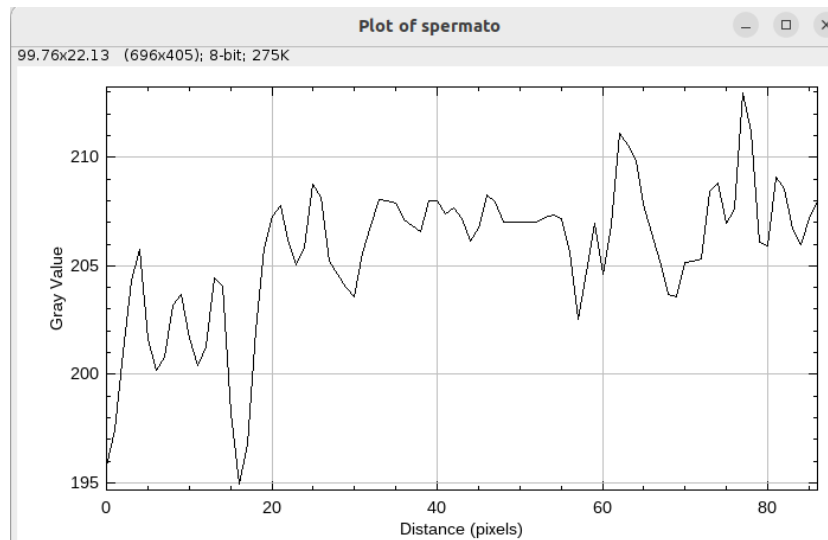
de l'image. Mais a-t-elle vraiment une couleur constante ? C'est ce que nous allons voir en examinant les profils 2D et 3D des images des lentilles et des spermatozoïdes.

### 3.3 L'importance de la couleur de l'arrière plan

Pour regarder si le fond de l'image des spermatozoïdes est constante (ou non), nous allons procéder de la même manière que précédemment. Si le fond est constant, alors la courbe du profil de la ligne sera constante.



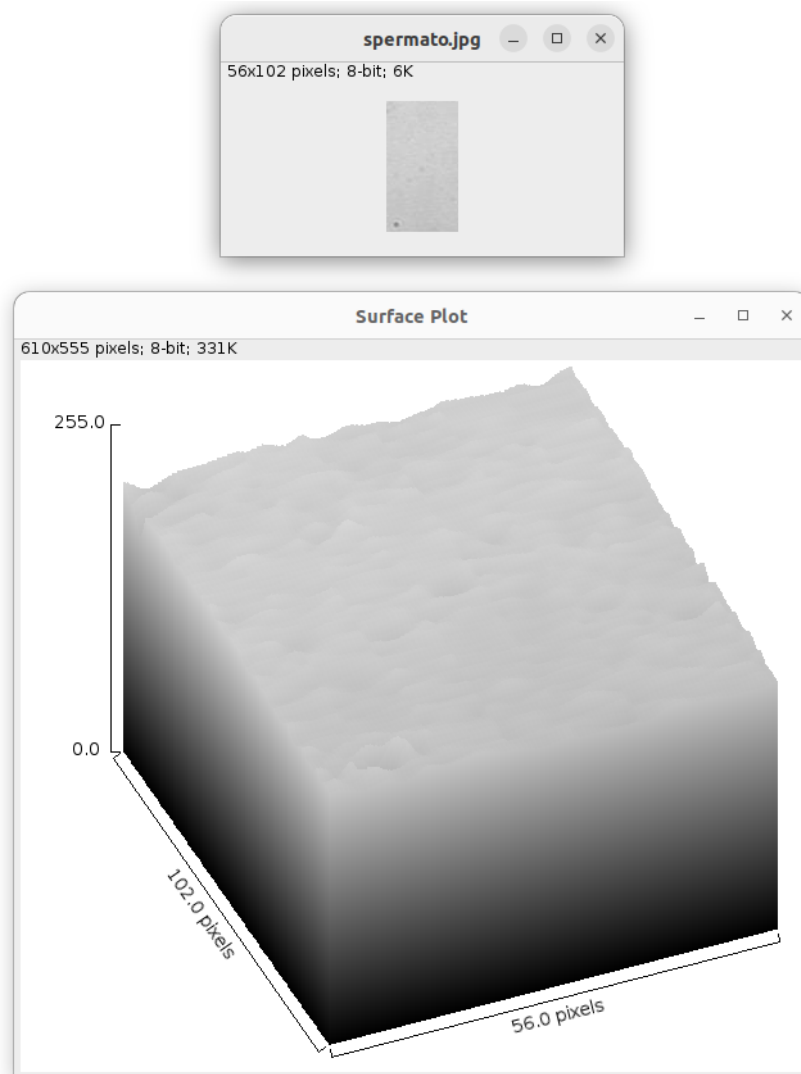
(1) Trait sélectionnée sur le fond de l'image des spermatozoïdes



(2) Profil de la ligne ci-dessus

La courbe du profil de la ligne n'est pas du tout constante : il y a des pixels de couleur noire et d'autres pixels de couleur blanche. Ainsi, nous pouvons en déduire que le fond n'a pas du tout un niveau de gris constant.

Vérifions avec le profil en 3D de la surface du fond (image ci-dessous). De la même manière, nous pouvons constater que la surface en 3D n'est pas lisse : il y a des grands écarts. Donc, nous pouvons en conclure, avec nos affirmations sur notre profil de ligne du fond de l'image des spermatozoïdes que **le fond n'a pas une couleur constante**.



Profil du fond de l'image des spermatozoïdes

Certaines images sont de mauvaises qualité et nécessitent un réglage de clarté et de contraste. Observons les différentes techniques pour modifier ces caractéristiques d'une image sur ImageJ.



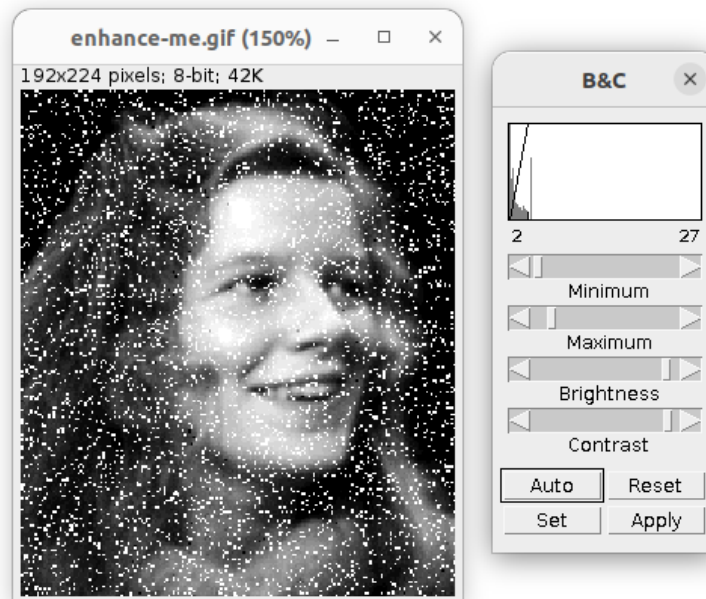
## 4 Clarté et contraste d'une image

L'image originale (montrée ci-dessous) que nous allons utiliser est de mauvaise qualité et a besoin de plusieurs rectifications sur son apparence.



Image d'origine

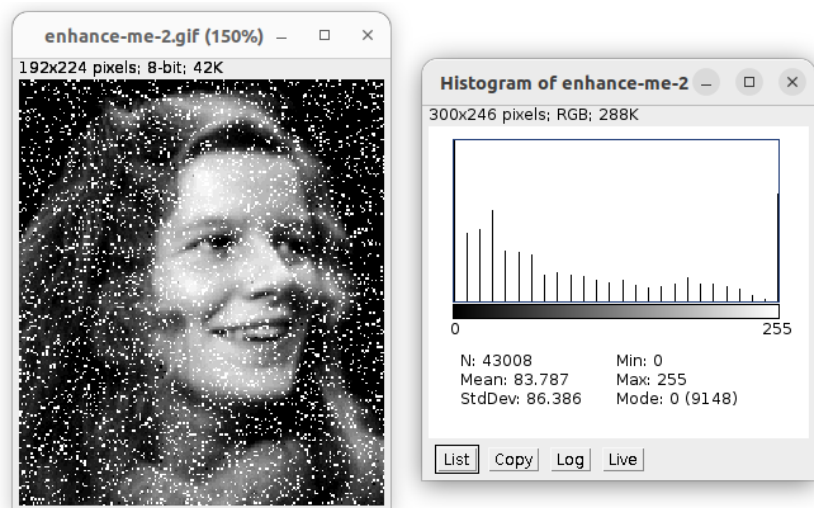
ImageJ possède trois techniques différentes pour régler les problèmes de clarté et de contraste. La première est l'outil **Brightness-Contrast** (Image / Adjust / Brightness-Contrast) qui modifie la luminosité et le contraste de l'image.



Les limites supérieures et inférieures des niveaux de gris peuvent être ajustées en modifiant les réglages minimum et maximum. La luminosité et le contraste de l'image peuvent être modifiés à l'aide des curseurs de luminosité et de contraste.

Le graphique présent juste au-dessus des curseurs a plusieurs caractéristiques : en abscisse, il y a les niveaux de gris et en ordonnée, il y a le nombre d'occurrence de pixels. Les deux chiffres sous le graphique sont les valeurs minimales et maximales de niveaux de gris des pixels affichés. Ces deux valeurs définissent la plage d'affichage. Les pixels dont la valeur est inférieure au minimum sont affichés en noir et ceux dont la valeur est supérieure au maximum sont affichés en blanc.

Le graphique linéaire, superposé à l'histogramme de l'image, montre comment les valeurs des pixels sont mises en correspondance avec les valeurs d'affichage sur huit bits.



Les deux autres techniques pour changer le contraste et la luminosité d'une image, implémentées dans ImageJ sont l'égalisation d'histogramme et la normalisation. Testons les deux outils sur l'image d'origine.

## 4.1 Égalisation d'histogramme

L'objectif de l'égalisation d'histogramme est une méthode d'ajustement du contraste qui utilise l'histogramme. Elle consiste à appliquer une transformation sur chaque pixel de l'image, et donc d'obtenir une nouvelle image à partir d'une opération indépendante sur chacun des pixels. Cette transformation est construite à partir de l'histogramme cumulé de l'image de départ. Il suffit de prendre la probabilité d'obtenir un pixel de valeur  $n$  cumulé et de le multiplier par la valeur maximale qu'on souhaite avoir (ici, **255**). Ainsi, la formule devient :

$$egalisation(n) = \sum_{i=0}^n P(i) * 255$$

L'outil **Equalize Histogram** (Process / EnhanceContrast / Equalize Histogram) de ImageJ nous permet d'avoir l'image égalisée sans faire les calculs. Ainsi, nous obtenons cette image et son histogramme.

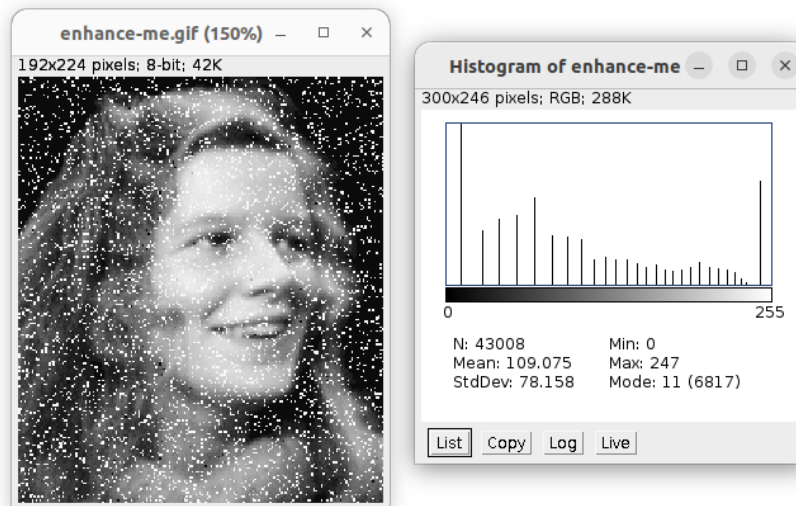


Image égalisée et histogramme

Entre l'image utilisant l'outil précédent et l'image égalisée, nous pouvons voir beaucoup de différences : la luminosité est beaucoup plus forte sur l'image égalisée et les niveaux de gris comprenant peu de valeurs sont tassés, alors que les niveaux de gris comprenant beaucoup de pixels sont étalés.

## 4.2 Normalisation d'histogramme

Normaliser l'histogramme d'une image consiste à appliquer une transformation d'histogramme à l'image afin d'étendre la plage de valeur de  $f$  à l'ensemble des valeurs disponibles. Concrètement, si l'on note  $fmin$  et  $fmax$  la valeur minimale et la valeur maximale de l'image, on souhaite trouver une transformation telle que la valeur minimale soit **0** et la valeur maximale soit **255**. La formule donne ainsi :

$$norm(x) = (x - fmin) * \frac{255}{fmax - fmin}$$

L'outil **Normalize** (Process / EnhanceContrast / Normalize) de ImageJ nous permet d'avoir l'image normalisée sans faire les calculs. Nous pouvons donc obtenir ces résultats-là :

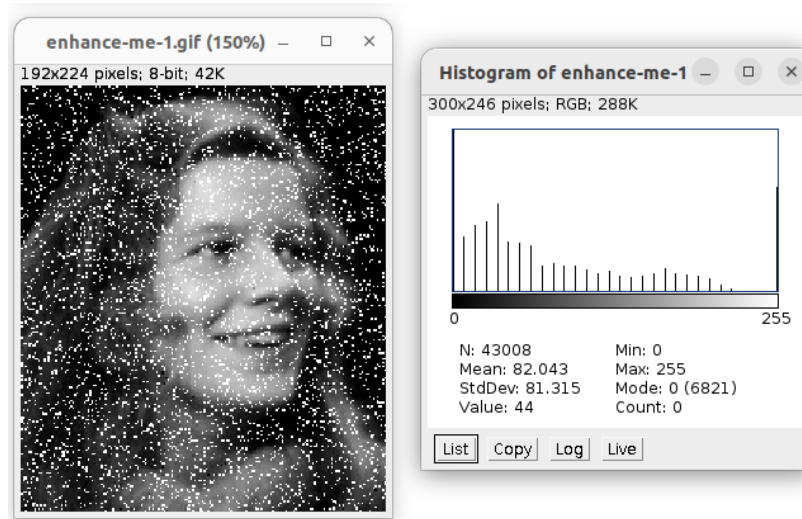


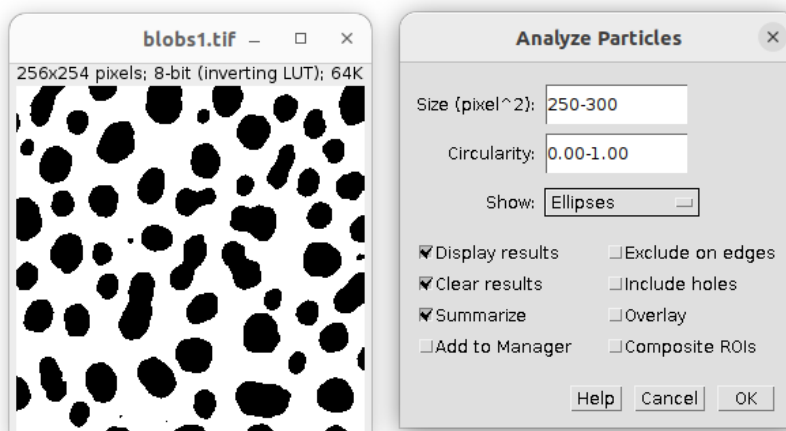
Image normalisée et histogramme

Sur l'image normalisée, la plage de valeur s'étend sur tout l'intervalle disponible et le contraste est plus léger que sur les deux images précédentes.

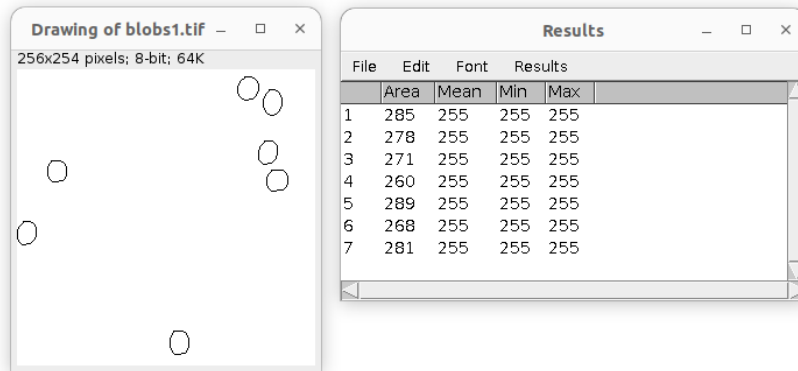
## 5 Analyse d'images binaires

Dans cette partie, on veut détecter le nombre d'éléments qui ont une surface particulière. Dans les parties précédents, nous avons vu que ImageJ est un logiciel permettant de séparer le fond et les RIO d'une image. Ici, nous avons une particularité différente, nous voulons extraire les données qui nous intéresse. Pour filtrer les différentes tailles des objets, il suffit de faire les mêmes opérations de la partie 1 et spécifier la surface que nous voulons dans le champ **Size**.

Sur l'image ci-dessous, nous voulons filtrer les objets ayant une surface de 250 pixels au carré à 300. Pour cela, il suffit d'utiliser l'outil **Tracing Tool** qui permet d'avoir une image seuillée. Ensuite, il suffit d'utiliser l'outil **Analyze Particles** et d'insérer **250-300** dans le champ **Size**.



(1) Image seuillée (2) Tableau de l'outil **Analyze Particles**



(3) Résultats

Sur les résultats ci-dessus, nous pouvons voir tous les blobs qui ont une surface comprise entre **250** pixels au carré à **300** pixels au carré. Il y en a **7**.

Maintenant que nous savons extraire des données et les filtrer, nous pouvons maintenant nous intéresser aux images en couleur RGB. Toutes les images en couleur sont composées de trois canaux de couleur (Rouge, Bleu et Vert). Comment pouvons-nous extraire certaines parties de l'image s'il y a plusieurs couleurs ?

## 6 Les images RGB

L'éclatement en trois images Rouge, Vert et Bleu peut se faire avec le logiciel. Par défaut, les tables de correspondance de niveaux de gris (LUT) indiquent la correspondance entre le niveau de gris réel dans une image et la couleur affectée à ce niveau de gris pour la visualisation.

Sur l'image ci-dessous, éclatons l'image du clown en trois images RGB. Trois nouvelles images vont apparaître.



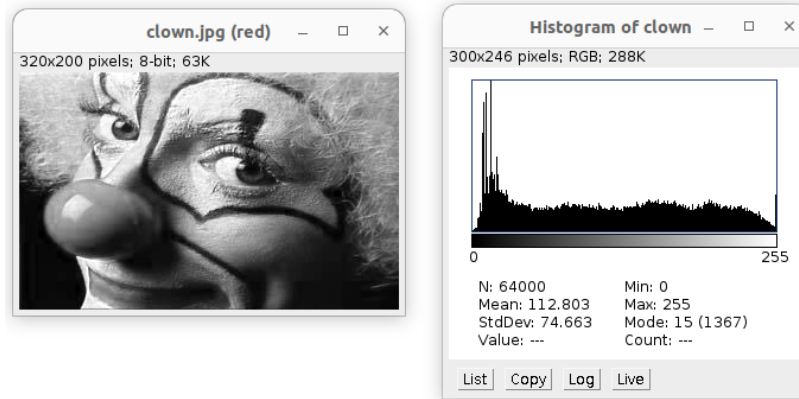
Image du clown

Pour cela, utilisons l'outil **Split Channels** (Image / Color / Split Channels).

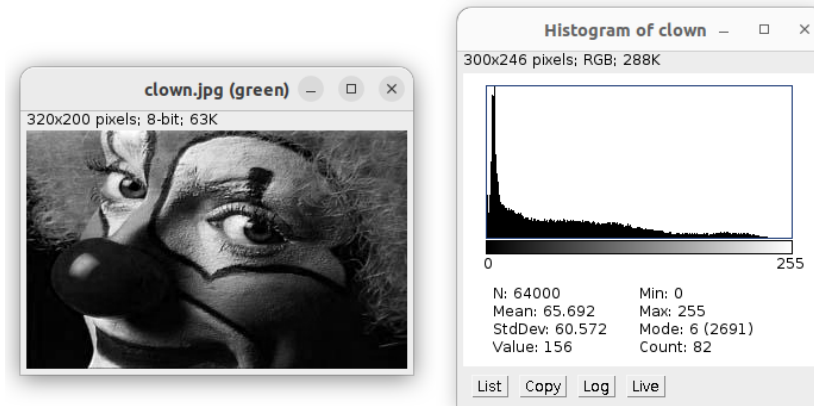


Images découpées du clown en Rouge, Vert et Bleu

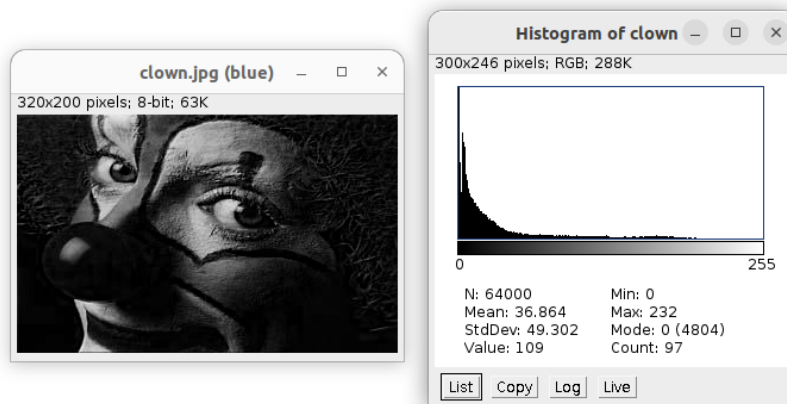
Nous pouvons remarquer que sur l'image d'origine, les couleurs dominantes sont le orange, le rouge et le jaune, qui sont des couleurs très proches du rouge. Sa luminosité est plus importante sur son image rouge. Regardons les histogrammes des trois images pour comprendre mieux.



(1) Histogramme de l'image Rouge



(2) Histogramme de l'image Vert



(3) Histogramme de l'image Bleu



Nous pouvons voir que la couleur la plus utilisée est la couleur Rouge, toutes les valeurs sont dispersées sur l'histogramme. C'est pour cette raison que la luminosité est plus importante. Comme la clarté est en lien direct avec l'utilisation des couleurs, nous allons essayer d'isoler l'une des trois couleurs sur l'image suivante.

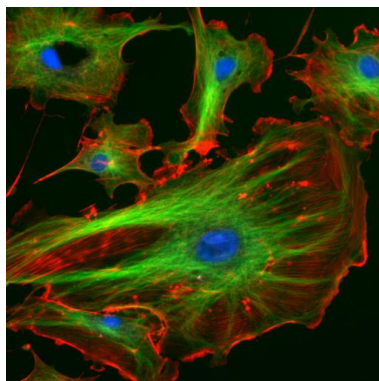
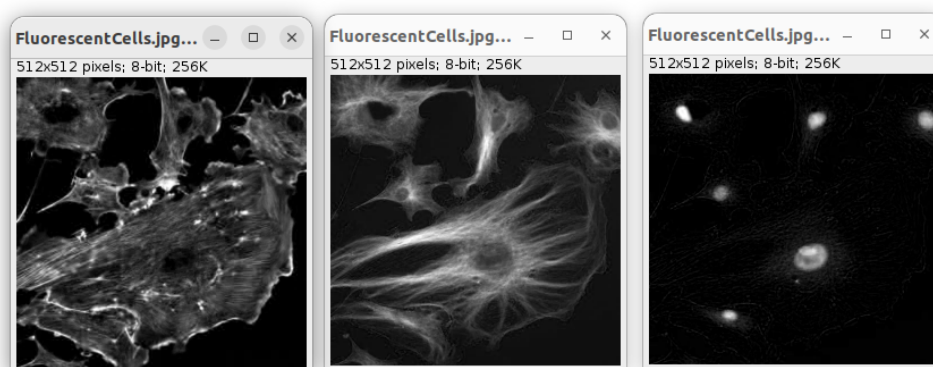


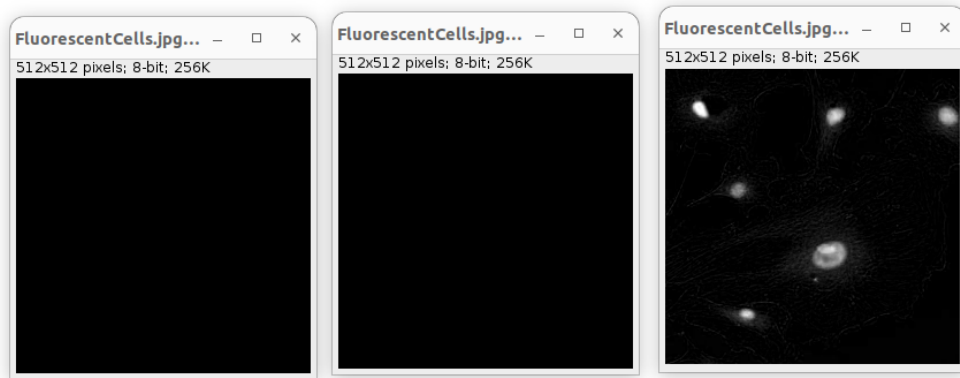
Image originale de cellules fluorescentes

Pour isoler l'une des trois couleurs, il faut d'abord séparer les trois couleurs.



Images découpées RGB

Nous avons vu juste précédemment que la clarté des images RGB influençait sur leur apparition sur l'image d'origine. Si nous voulons isoler la couleur bleu, il faut faire en sorte que les couleurs Rouge et Vert ne s'affichent plus. Pour cela, changeons la clarté de ces deux images et mettons les en noir.



Images R et G en noir

Ensuite, regroupons les images avec l'outil **Merge Channels** (Image / Color / Merge Channels).



Images R et G en noir

Et voilà !

## 7 Le filtrage



(1) Utilisation de **Smooth**



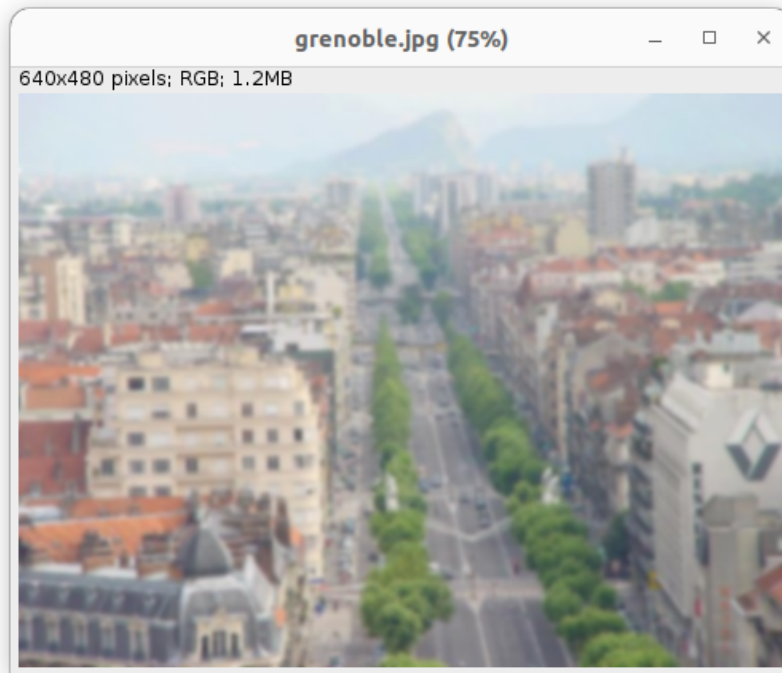
## (2) Utilisation de **Sharpen**

L'outil **Sharpen** (Process / Sharpen) augmente le contraste et accentue les détails de l'image, mais peut également accentuer le bruit. Ce filtre utilise les facteurs de pondération suivants pour remplacer chaque pixel par une moyenne pondérée du voisinage 3x3.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 12 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Quant à lui, l'outil **Smooth** (Process / Smooth) rend floue l'image ou la sélection active. Ce filtre remplace chaque pixel par la moyenne de son voisinage 3x3.

ImageJ possède aussi la fonctionnalité d'appliquer un filtre gaussien sur une image. C'est l'outil **Gaussien Blur** (Process / Filters / Gaussien Blur).

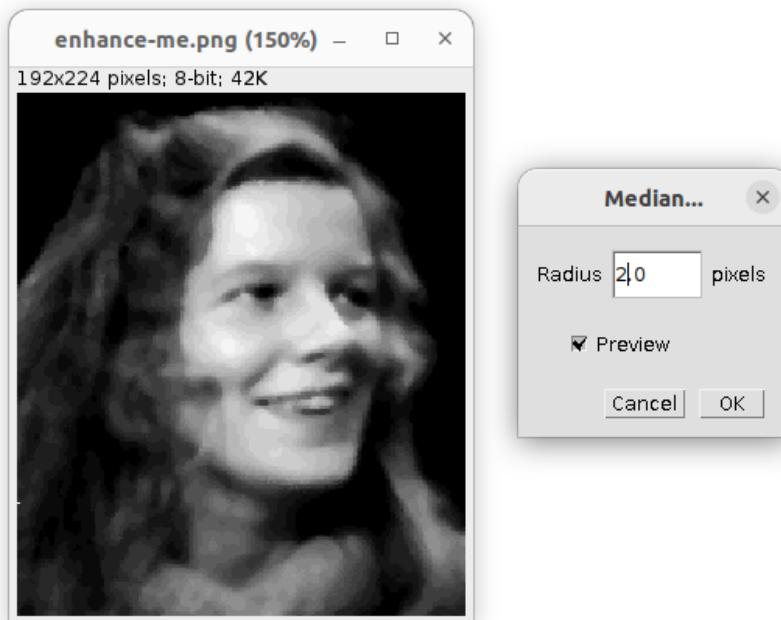


Application d'un filtre gaussien avec un masque de taille 2

Le filtre gaussien donne un meilleur lissage et une meilleure réduction du bruit que le filtre moyennneur.

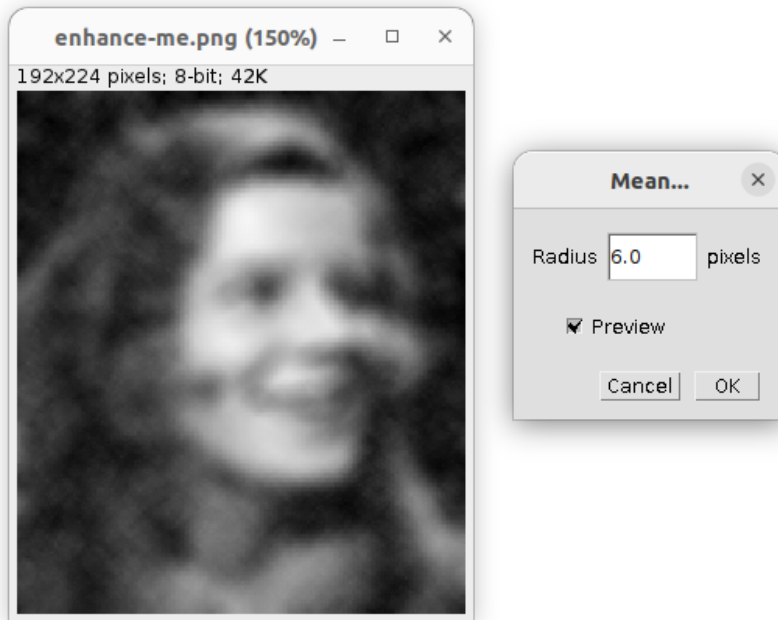
### 7.1 Différence entre le filtre moyennneur et le filtre médien

Le filtre médian est un filtre numérique non linéaire, souvent utilisé pour la réduction de bruit tout en conservant les contours de l'image. Pour l'utiliser, il suffit d'utiliser l'outil **Median** (Process / Filters / Median) sur l'image souhaitée.



Sortie du filtre médian

Les filtres moyennisseurs, calculent la moyenne, éventuellement pondérée, des pixels situés dans le voisinage de chaque pixel. Cette famille de filtres permet de réduire le bruit dans l'image, ce qui rend les zones homogènes plus lisses. Par contre, les contours sont fortement dégradés, et les structures trop fines peuvent devenir moins visibles. Pour l'utiliser, il suffit juste d'utiliser l'outil **Mean** (Process / Filters / Mean) sur l'image souhaitée.



Sortie du filtre moyeneur

Le filtre médian permet d'éliminer les valeurs aberrantes sans se limiter à faire un calcul de moyenne qui aura tendance à contaminer les valeurs voisines avec cette valeur aberrante et flouter l'image. De ce fait, pour cette image, le filtre médian est bien meilleur, de taille 2.

En général, la sortie d'un filtre est la convolution de l'image. Cette convolution donne un algorithme de calcul d'un filtre. Cette convolution peut être également calculée des techniques récursives. Après avoir vu les différents types de filtre, retrouvons l'algorithme qui permet de procéder à un filtre moyeneur.

## 8 Convolution

Pour cette dernière partie, nous allons utiliser l'outil **Concuvres** (Process / Filters / Convulves). Dans le cas d'un filtre de lissage comme le filtre moyeneur, il permet de remplacer le niveau de gris d'un pixel par la moyenne des niveaux des pixels voisins avec le filtre suivant :



$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Utilisons le sur l'image de grenoble et regardons le résultat :



Image de grenoble avec un filtre moyennneur de taille 3 \* 3

Le filtre de lissage laisse du flou apparaître.