PSE : Peigner du collagène pour faire du muscle

SEGURET Magali
REY Baptiste
CHOMETON Ronan

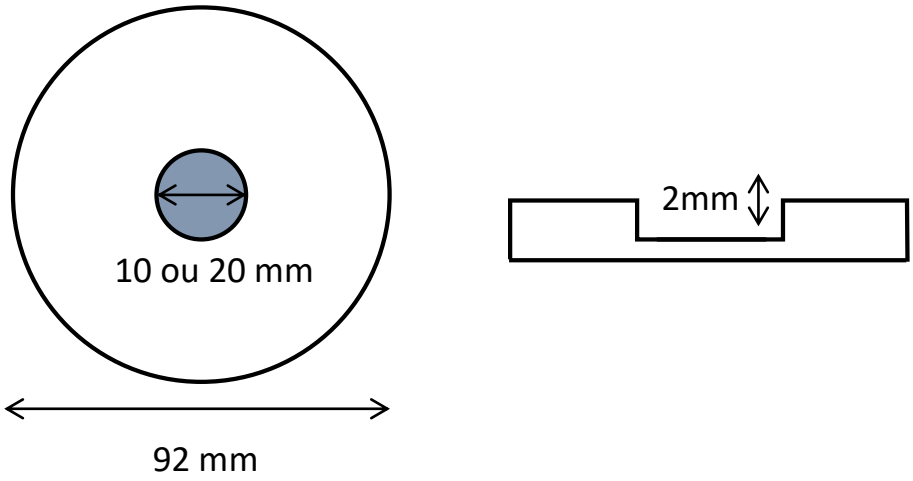
Protocoles et méthodes

I- Réalisation du gel de collagène

a) Réalisation des puits de collagène en PDMS

Produits	Matériel
- PDMS et réticulant	- Une boîte de Pétri - Plexiglas de 2 mm d'épaisseur

Pour pouvoir appliquer des contraintes sur notre collagène, nous utilisons des puits en PDMS tels que celui représenté ci-dessous.



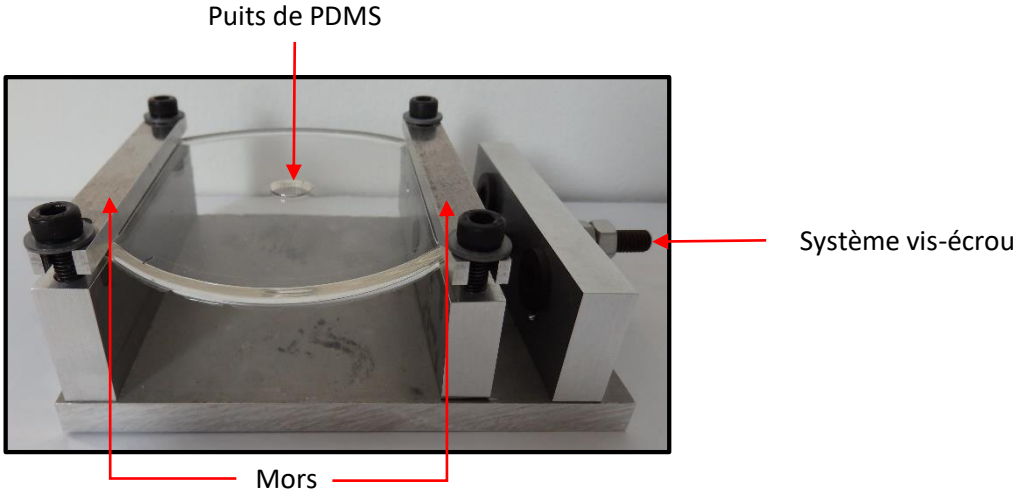
Ces puits sont réalisés à l'aide d'un moule formé d'un couvercle de boîte de Pétri, à l'intérieur duquel est fixée une pièce circulaire de plexiglas de 2 mm d'épaisseur à la colle forte. Deux diamètres de puits ont été testés au cours des expériences : 20 mm puis 10 mm.

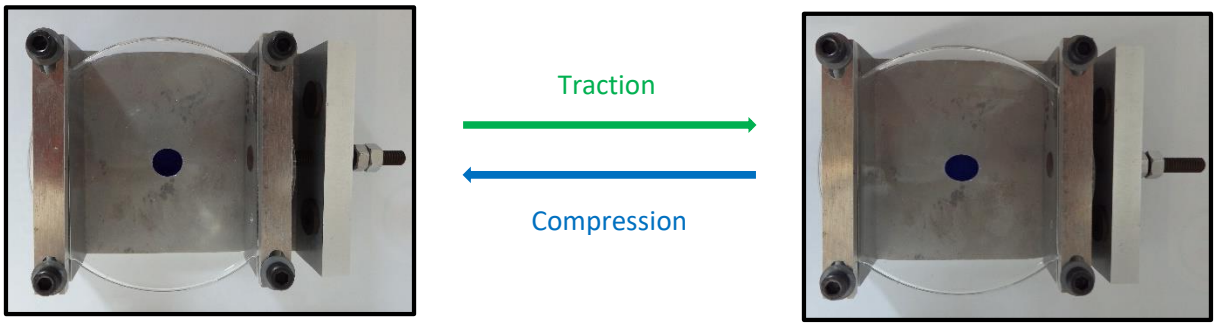
Protocole : on réalise un mélange PDMS/réticulant 93/7 en masse. Celui-ci est placé sous vide à plusieurs reprises jusqu'à ce que l'ensemble des bulles soient éliminées. Il est versé dans le moule décrit préalablement jusqu'à une hauteur d'environ 5 mm. La réticulation a lieu dans une étuve à 70°C pendant 2h.

b) Polymérisation et alignement du collagène

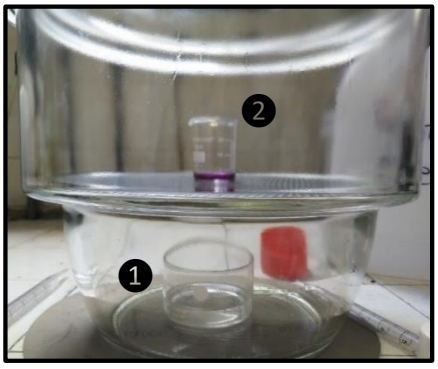
Produits	Matériel
- Solution de collagène de type I (issu de queue de rat) à 7 mg/mL dans l'acide acétique 0.2N - Solution d'hydroxyde d'ammonium à 28.0-30% NH_3 base - Alizarine 97% - Acide acétique 0.2N - Phosphate Buffured Saline à	- Machine de traction réalisée sur mesure à l'atelier de l'ESPCI Paris - Dessiccateur de diamètre 25cm - Deux structures PDMS comme décrite ci-dessus

Machine de traction :

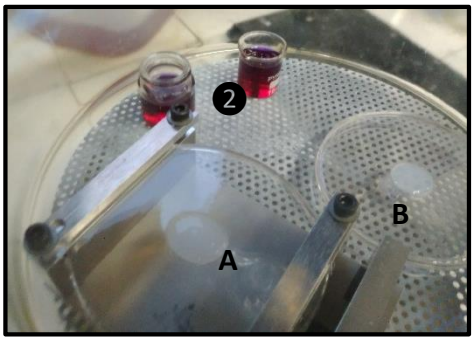




Protocole : Dans le dessiccateur, on place un cristallisoir contenant environ 50mL de $NH_4OH_{(aq)}$ (❶), un bécher contenant une pointe de spatule d'alizarine dissous dans environ 10mL d'acide acétique 0,2N (❷). Le dessiccateur est fermé afin de créer l'atmosphère saturée en ammoniac. On vérifie l'augmentation de pH grâce au changement de couleur de l'alizarine qui passe du jaune au rouge-violet.



A l'aide d'une pipette P1000 on dépose au fond de chaque puits 1mL de solution de collagène. On place un des 2 puits, que l'on appellera puits A, dans la machine de traction et l'on étire le PDMS de 10mm (10,8% d'extension). On place ensuite la machine de traction et le puits de collagène témoin, puits B, dans le dessiccateur. On laisse polymériser pendant 2h.



On vérifie que le collagène a bien polymérisé : il doit être sous la forme d'un gel, donc ne pas couler, et avoir une couleur blanchâtre. On sort alors le puits A, toujours dans la machine de traction étirée, et le puits B du dessiccateur. Pour le puits A, on repère à l'aide d'un marqueur l'axe de traction. On relâche ensuite la traction de sorte que le puits reprenne sa forme initiale. Le collagène gélifié est donc comprimé dans la direction de la traction. Si les expériences suivantes nécessitent de démouler les galettes de collagène, il faut attendre 1h après la re-compression du gel avant de ce faire.

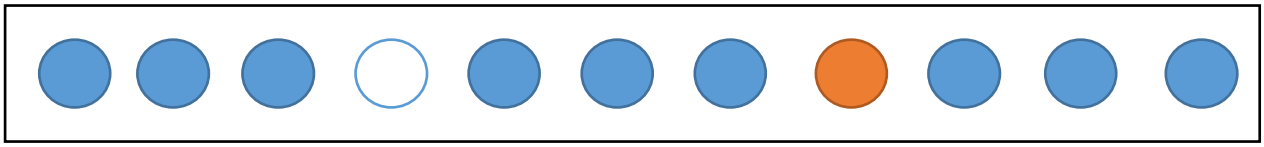
II- Observation de l'alignement des fibres

a) Observation des fibres au microscope à fluorescence

Produits	Matériel
- Phosphate Buffered Saline - Anticorps primaire : Collagen I Monoclonal Antibody - Anticorps secondaire : Alexa Fluor 488 Anti-Mouse – Référence A11001	- Plaque 96 puits - Pipettes P1000, P200, P20, P2 - Spatule de largeur adaptée aux puits

Préparation de l'échantillon – fixation : On découpe, dans le gel contraint et le gel témoin, un échantillon de collagène au scalpel. Il est difficile de contrôler la taille de ces échantillons, ils doivent avoir une longueur caractéristique de 1mm. Sur une plaque de 96 puits on prépare une ligne avec les solutions de lavages successifs dans l'ordre suivant.

PBS, Solutions d'anticorps I, **Solutions d'anticorps II**



Solution d'anticorps I : 1/1000 (v/v) d'anticorps I dans le PBS

Solution d'anticorps II : 1/500 (v/v) d'anticorps II dans le PBS

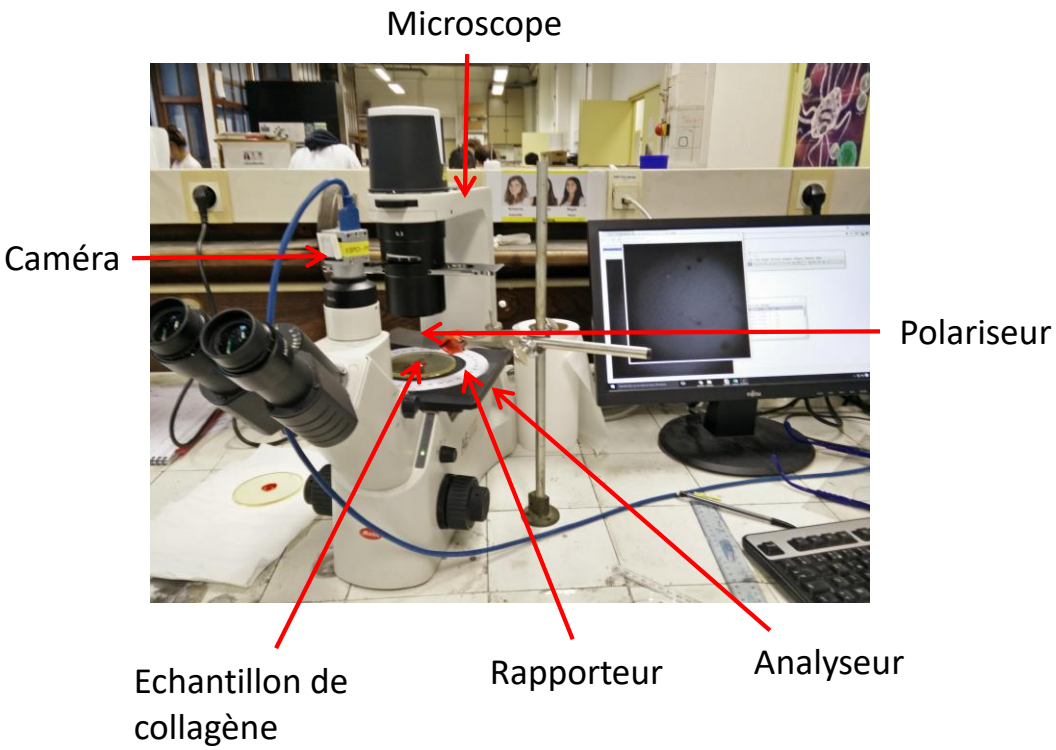
On réalise les 3 lavages au PBS, puis on laisse une heure dans la solution d'anticorps I. On refait 3 lavages au PBS puis on laisse 30 minutes dans l'anticorps secondaire. On termine par 3 lavages au PBS.

Observation au microscope à fluorescence (marque Zeiss) : On dépose chaque échantillon sur une lame que l'on recouvre d'une lamelle. On choisit la longueur d'onde d'excitation à 488 nm, on a une émission à 525nm.

b) Observation de l'alignement des fibres

Produits	Matériel
- Galettes de collagène non démoulées	- Un microscope optique muni d'une caméra reliée à PythonViewer - Logiciel ImageJ - Deux polariseurs - Un rapporteur circulaire adaptable au microscope

Protocole : On place l'échantillon à analyser sur le plateau du microscope, et l'on ajoute un polariseur et un analyseur croisés de part et d'autre du plateau. On munit le plateau d'un rapporteur. **Réalisation du blanc :** on enregistre une image avec le PDMS seul, sans collagène. On fait tourner le plateau sur lequel est positionné l'échantillon de collagène, par pas de 10°, et une image est enregistrée via la caméra. On mesure ensuite l'intensité moyenne des images via ImageJ.



III- Quantification de l'alignement des fibres

a) Préparation de l'échantillon

Produits	Matériel
- PFA 8% (masse) - Phosphate Buffered Saline	

Afin de fixer la structure, on immerge le gel dans un bain de PFA 8% en masse dans le PBS pendant plusieurs heures. On lave dans plusieurs bains de PBS. Ceci permet d'avoir des échantillons dont la structure est fixée et qui garderont leur forme même en séchant. Les échantillons sont ensuite observés au MEB. Préalablement, une portion de l'échantillon est découpée au scalpel dans lequel est pratiqué une incision. Le but est d'ôter la couche superficielle qui ne présente pas la structure de réseau fibrillaire. On peut ensuite observer une couche de gel, où apparaissent les fibres de collagène.

b) Analyse numérique via un programme Python

La quantification des orientations des fibres de collagène nécessite une détection des bords suivie d'une extraction de lignes par transformation de Hough.

1. Détection des bords

Le principe à la base de la détection des bords sur les images de microscopie repose sur la dérivation du premier ordre. Un bord est une variation importante du niveau de gris, donc un gradient élevé, localisé sur une région de l'image. Le traitement de l'image repose sur la méthode de Canny et comprend quatre étapes :

- Lissage Gaussien : filtre passe-bas pour s'affranchir du puits
- Différentiation de Sobel
- Seuillage
- Suppression des non-maxima

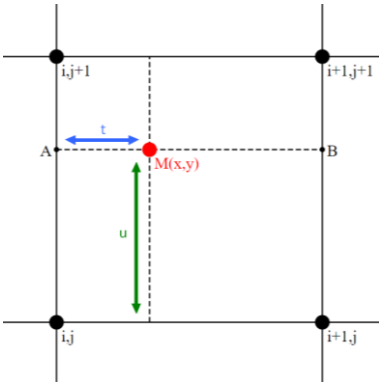
Différentiation de Sobel : l'opérateur de Sobel calcule le gradient de l'intensité de chaque pixel, soit le vecteur à deux dimensions comprenant les dérivées selon les directions horizontales et verticales. Soit $\boldsymbol{M}$ l'image source, $\boldsymbol{G_x}$ et $\boldsymbol{G_y}$ sont les deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point.

$$\boldsymbol{G_x} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} * \boldsymbol{M} \qquad \boldsymbol{G_y} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} * \boldsymbol{M}$$

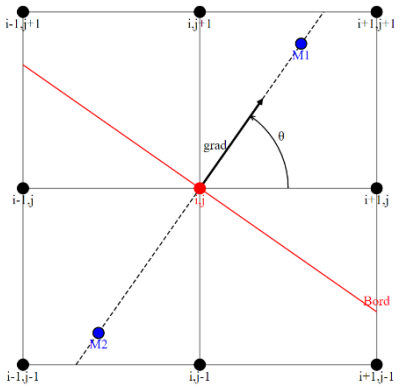
La norme du gradient est donc donnée par $\boldsymbol{G} = \sqrt{\boldsymbol{G_x}^2 + \boldsymbol{G_y}^2}$ et sa direction par $\boldsymbol{\Theta} = atan2(\boldsymbol{G_y}, \boldsymbol{G_x})$.
Le tracé de la norme correspond aux contours de l'image.

Seuillage : Afin de s'affranchir du bruit, un seuil est établi pour la valeur du gradient en un pixel. Les pixels étant en dessous de ce seuil sont éliminés.

Suppression des non-maxima : les bords ont une largeur de plusieurs pixels, cette largeur dépendant de l'image et de l'intensité du lissage. Pour déterminer si un pixel est un maximum de gradient, on utilise une interpolation bilinéaire à 4 pixels. Cela permet de connaître la valeur du gradient en un point qui ne coïncide pas avec un pixel.



On connait la valeur d'un point M(x,y) par la formule suivante : on définit les paramètres $t = y - j$ et $u = x - i$. La valeur du gradient au point A et B est donnée par $V_A = (1-u) * V_{i,j} + u * V_{i,j+1}$ et $V_B = (1-u) * V_{i+1,j} + u * V_{i+1,j+1}$. La valeur du gradient au point M est donnée par $V_M = (1-t) * V_A + t * V_B$.



La figure suivante représente un pixel (i,j) et ses 8 proches voisins. Le bord en ce point (trait plein) est perpendiculaire au gradient. On a aussi représenté deux points M_1 et M_2 situés sur la normale (trait pointillé), de part et d'autre du bord, à une distance unité du point (i,j). La valeur de la norme du gradient $\boldsymbol{G}$ aux points M_1 et M_2 est calculée par interpolation bilinéaire à partir des 4 pixels voisins. Le pixel (i,j) est retenu si son gradient est supérieur à ceux des points M_1 et M_2. Dans le cas contraire, le pixel est éliminé. On élimine ainsi les pixels qui ne sont pas sur un maximum du gradient (maximum le long de la normale).

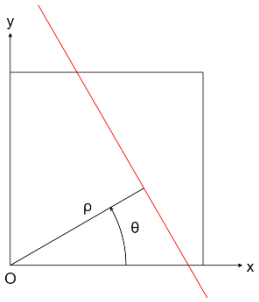
Ci-dessous le programme de détection du contour :

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 11 14:15:00 2018

@author: Ronan
"""

"""Programme de détection des contours d'une image"""

import numpy as np #listes
import math #fonctions pour nombres réels
import cmath #fonctions pour nombres complexes


from scipy.ndimage.filters import convolve,gaussian_filter #scipy.ndimage : traitement d'images
from scipy.misc import imsave #fonctions diverses / sauvegarder image en tableau gris ou RGB
from  matplotlib.pyplot import * #librairie pour les graphes
from PIL import Image

"""Récuperation de l'image à traiter"""
im = Image.open('temoin.tif') #tableau
img = np.array(im)
M=img*1.0

"""Lissage gaussien"""
M = gaussian_filter(M,7) #écart-type variable

"""Differentiation de Sobel"""
sobelX = np.array([[-1,0,1],[-2,0,2],[-1,0,1]]) #matrice de Sobel horizontale
sobelY = np.array([[-1,-2,-1],[0,0,0],[1,2,1]]) #matrice de Sobel verticale
Gx = convolve(M,sobelX) #dérivée horizontale approximée
Gy = convolve(M,sobelY) #dérivée verticale approximée
gradient = Gx+Gy*1j #matrice du gradient
G = np.absolute(gradient) #norme du gradient
theta = np.angle(gradient) #direction du gradient

"""Seuillage"""
seuil = 12.5 #fixation du seuil
s = G.shape #dimensions de la matrice gradient
for i in range(s[0]):
    for j in range(s[1]):
        if G[i][j]<seuil:
            G[i][j] = 0.0 #élimination des des pixels sous le seuil
```

```python
Gmax = G.copy() #copie de la matrice gradient

"""Interpolation bilinéaire à 4 pixels"""
def interpolation(M,x,y): #fonction d'interpolation
    s = M.shape #dimensions matrice image
    i = math.floor(x) #partie entière de x
    j = math.floor(y) #partie entière de y
    t = x-i #paramètre horizontal
    u = y-j #paramètre vertical
    u1 = 1.0-u
    t1 = 1.0-t
    if j==s[0]-1: #le point est sur le bord vertical
        if i==s[1]-1: #le point est aussi sur le bord horizontal donc il est dans le coin
            return M[j][i]
        return u*M[j][i]+u1*M[j][i+1]
    if i==s[1]-1: #le point est uniquement sur le bord horizontal
        return t*M[j][i]+t1*M[j+1][i] #le point est uniquement sur le bord vertical

    return t1*u1*M[j][i] + t*u1*M[j][i+1]+ t*u*M[j+1][i+1] + t1*u*M[j+1][i] #le point n'est pas sur un bord

for i in range(1,s[1]-1): #l'ensemble des pixels sont analysés
    for j in range(1,s[0]-1):
        if G[j][i]!=0: #on est sur un bord
            cos = math.cos(theta[j][i]) #orientation du gradient
            sin = math.sin(theta[j][i])
            g1 = interpolation(G,i+cos,j+sin) #valeur du gradient en M1
            g2 = interpolation(G,i-cos,j-sin) #valeur du graident en M2
            if (G[j][i]<g1) or (G[j][i]<g2): #condition de non maximum
                Gmax[j][i] = 0.0 #suppression du non maximum

"""Seuillage"""
Gfinal = Gmax.copy()
seuil = 13.5 #fixation du seuil
for j in range(s[0]): #l'ensemble de l'image est parcourue
    for i in range(s[1]):
        if Gfinal[j][i]<seuil: #binarisation de l'image
            Gfinal[j][i] = 0.0
        else:
            Gfinal[j][i] = 255.0

Gfinal = Gfinal[0:1750][:] #sélection d'une portion de l'image
np.save('contour_temoin.npy',Gfinal) #sauvegarde de la matrice des contours
imsave('contour_temoin.png',Gfinal, cmap=cm.gray) #sauvegarde de l'image des contours
```

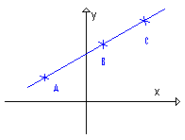
2. Extraction de lignes

Une fois les contours détectés sur l'image, il est intéressant d'en extraire les lignes droites. C'est la tâche de l'algorithme de Hough. La transformée de Hough consiste à représenter chaque point de contour détecté dans un espace de paramètres à deux dimensions :



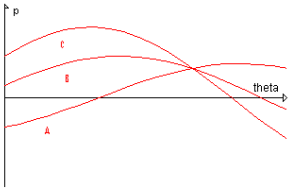
- Une droite est caractérisée par deux paramètres, elle est donc représentée par un point dans cet espace de paramètres $(\theta; \rho)$ où θ est l'angle que fait la normale à la droite avec l'axe des abscisses et ρ est la longueur du segment perpendiculaire à la droite passant par l'origine
- Si l'on considère l'ensemble des droites passant par un point, l'image de cet ensemble est une courbe dans l'espace de paramètres

Image originale



La transformée de Hough d'un point de l'image analysée est la courbe de l'espace des paramètres correspondant à l'ensemble des droites passant par ce point. Si des points sont colinéaires, alors toutes les courbes de l'espace de paramètres se coupent au point représentant la droite en question.

Espace de Hough



Si on note N_x et N_y les nombres de pixels horizontalement et verticalement, les domaines sont : $\rho \in \left[0, N_x^2 + N_y^2\right]$ et $\theta \in [0, \pi]$. L'accumulateur est une matrice qui correspond à une discrétisation de ce domaine. On note N_θ et N_ρ les nombres de points de l'accumulateur dans ses deux dimensions. Initialement, l'accumulateur est vide. Pour chaque pixel $M(x, y)$ de l'image, on incrémente dans l'accumulateur les points de la courbe d'équation $\rho = x * cos\theta + y * sin\theta$. Il s'agit d'une sinusoïde. Il faut bien sûr échantillonner cette courbe de manière à remplir l'accumulateur de manière assez dense.

En fonction des indices (i, j) des pixels, l'équation est $\rho = i * cos\theta + \left(N_y - j\right) * sin\theta$. Lorsque tous les points de l'image ont été traités, on sélectionne les éléments de l'accumulateur remplis au-dessus d'un certain seuil, qui sont alors interprétés comme les paramètres de droites correspondant à des lignes droites sur l'image. Le nombre d'éléments de l'accumulateur doit être choisi en fonction de la précision souhaitée.

Afin d'observer la répartition de l'orientation des fibres, l'accumulateur est représenté selon la direction θ.

Ci-dessous le programme de la transformée de Hough :

```
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 12 14:44:17 2018

@author: Ronan
"""

"""Détection de droites par transformée de Hough"""

import numpy as np #listes
import math #fonctions pour nombres réels
import cmath #fonctions pour nombres complexes
import  matplotlib.pyplot as plt #librairie pour les graphes
from PIL import Image #Python Image Library

M = np.load("contour_temoin.npy") #chargement du la matrice du contour

"""Zonage en cercle du contour"""
(Ny,Nx) = np.shape(M) #dimensions de l'image
i_centre=Nx/2 #coordonnées du centre de l'image
j_centre=Ny/2
rayon=750 #rayon du cercle d'analyse
r2=rayon**2
for j in range(Ny):
    for i in range(Nx):
        check=(i-i_centre)**2+(j-j_centre)**2
        if check > r2: #pixel en dehors du cercle
            M[j][i] = 0

"""Définition de l'accumulateur de Hough"""
rho = 1.0 #résolution en distance
theta = 1.0 #résolution en angle
Ntheta = int(180.0/theta) #nombre de points en angle
Nrho = int(math.floor(math.sqrt(Nx*Nx+Ny*Ny))/rho) #nombre de points en distance
dtheta = math.pi/Ntheta #petit élément d'angle en radians
drho = math.floor(math.sqrt(Nx*Nx+Ny*Ny))/Nrho #petit élément de distance
accum = np.zeros((Ntheta,Nrho)) #matrice de l'accumulateur vide

"""Remplissage de l'accumulateur"""
for j in range(Ny):
    for i in range(Nx):
        if M[j][i]!=0:#points du contour
            for i_theta in range(Ntheta): #parcours l'ensemble des directions des droites passant par le pixel (i,j)
```

```
            for i_theta in range(Ntheta): #parcours l'ensemble des directions des droites passant par le pixel (i,j)
                theta = i_theta*dtheta #calcul de theta en radians
                rho = i*math.cos(theta)+(Ny-j)*math.sin(theta) #calul de rho
                i_rho = int(rho/drho) #incrémentation de rho
                if (i_rho>0) and (i_rho<Nrho): #i_rho est bien dans l'accumulateur
                    accum[i_theta][i_rho] += 1 #incrémentation

"""Seuillage"""
seuil=70 #fixation du seuil
accum_seuil = accum.copy() #copie de l'accumulateur
for i_theta in range(Ntheta):
    for i_rho in range(Nrho):
        if accum[i_theta][i_rho]<seuil:
            accum_seuil[i_theta][i_rho] = 0 #élimination des valeurs en dessous du seuil
angle = [i*dtheta*180/math.pi for i in range(Ntheta)] #matrice des angles discrétisés
new_accum = np.sum(accum_seuil,axis=1) #accumulateur selon la direction théta
new_accum[45]=(new_accum[44]+new_accum[46])/2
surface = np.sum(new_accum) #aire sous la courbe
new_accum = new_accum/surface #accumulateur normalisé par son intégrale
plt.plot(angle,new_accum) #graphe de l'accumulateur en fonction de théta
plt.xlabel('Orientation des fibres (°)')
plt.ylabel('Occurence')
plt.title("Profil d'orientation des fibres dans le gel comprimé")
plt.show()
plt.savefig('orientation_comprime.png', bbox_inches='tight')
```

IV- Différenciation des cellules sur le collagène

On cultive des cellules sur des pastilles de collagène contraint et témoin et on observe leur forme et position au microscope à fluorescence, après une semaine de mise en culture.

a) Culture des cellules

On utilise des cellules C2C12, des myocytes de souris. Elles sont déposées sur les gels dans une boîte de Pétri et le tout est immergé dans du milieu de culture.

b) Fixation et coloration

On fixe les cellules et le collagène dans le PFA pendant 1h. On perméabilise ensuite les cellules afin de permettre la coloration par les anticorps. Pour cela, on baigne les échantillons dans du triton-X 0,2% (v/v) dans le PBS pendant 20 min. Enfin, on colore les noyaux et les filaments d'actine, qui donnent la structure de la cellule. On utilise des solutions à 1 µmol/L de SIR-actin (actine) et de SYTO Orange fluorescent Nucleic Acid (noyaux) dans 2,5 mL de PBS, quantité suffisante pour que les échantillons soient bien recouverts. On les laisse 1h en solution.

c) Observation

On observe au microscope à fluorescence (marque Zeiss) en utilisant les filtres adéquats.