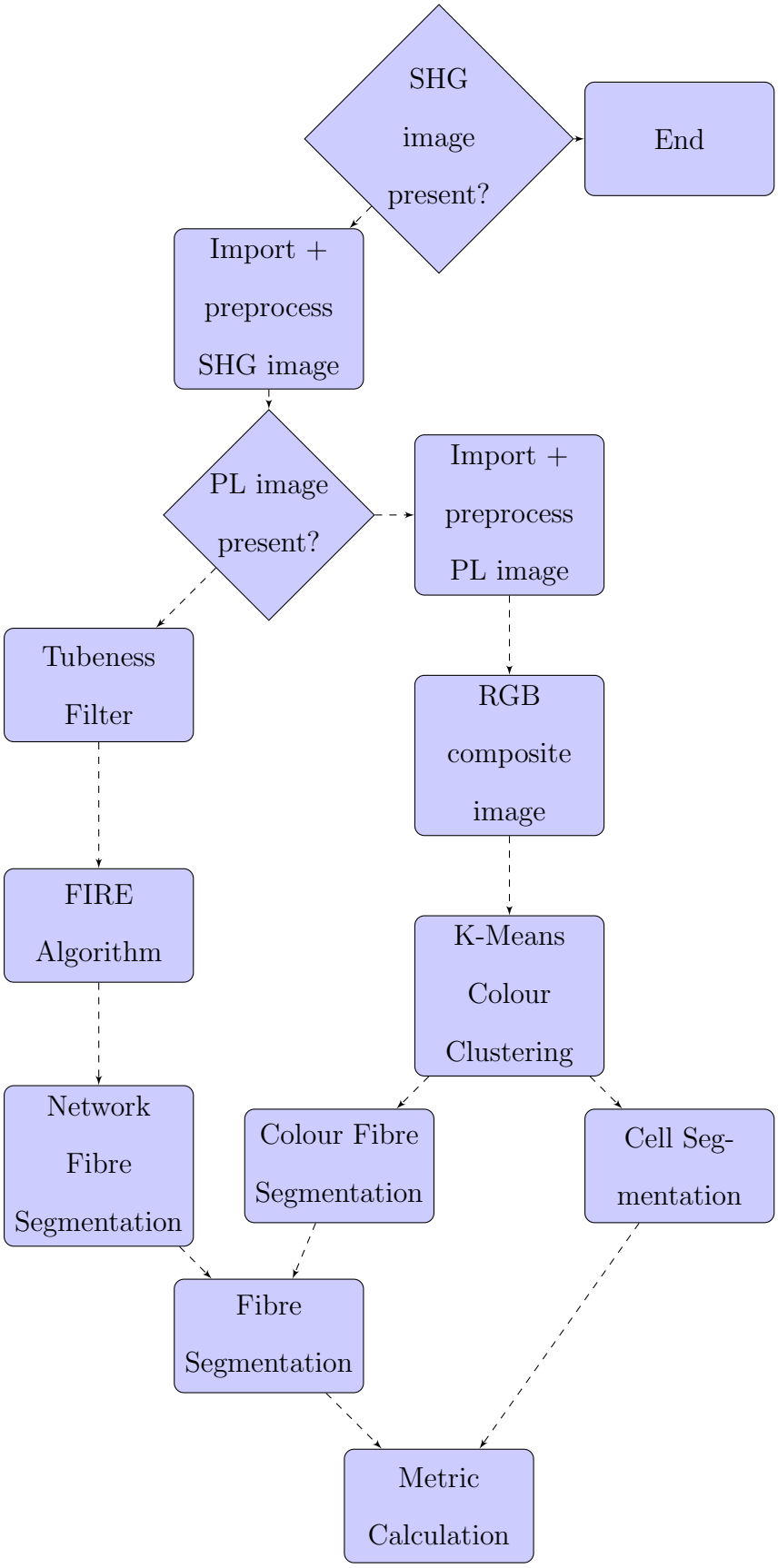
PyFibre - Python Fibrous Image Analysis Toolkit

Francis G.J. Longford

April 1, 2019

PyFibre (Python Fibrous Image Analysis Toolkit) is a computer program written in Python purposely built to analyse fibrous tissue. It is designed to be as automated as possible so that it can easily can be applied to multiple image sets in order to generate databases of properties for further large-scale analysis.

1 Flowchart


SHG image present?
End
Import + preprocess SHG image
PL image present?
Import + preprocess PL image
Tubeness Filter
RGB composite image
FIRE Algorithm
K-Means Colour Clustering
Network Fibre Segmentation
Colour Fibre Segmentation
Cell Segmentation
Fibre Segmentation
Metric Calculation

2 Noise Removal

Each image goes through a pre-processing stage in order to remove noise and enhance any significant features present. This is crucial for successful and efficient implementation of the FIRE algorithm later on.

2.1 Noise Processes

There are many different types noise in images. Some common examples include white noise, shot noise and salt-and-pepper noise[1]. Each are generated by different processes and so display different probability distributions for pixel intensities. Therefore it is possible that separate methods will be required in order to remove or reduce their influence in an image.

2.1.1 White Noise

White noise is generated during image acquisition by random thermal fluctuations and therefore demonstrates a Gaussian distribution of pixel intensities, with a mean μ and standard deviation σ.

$$P(I) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(I-\mu)^2}{2\sigma^2}\right] \tag{1}$$

This can be initially reduced by using higher camera illumination or averaging of multiple images over a longer exposure. However, there are also many algorithms that can effectively smooth images displaying high levels of white noise at the loss of contrast though Gaussian convolution.

2.1.2 Shot Noise

Shot noise is also generated during image acquisition by quantum fluctuations in photons and therefore demonstrates a Poisson distribution of pixel intensities, with mean photon count λ. However, for high pixel image intensities (and therefore large values of λ) shot noise approximates a Gaussian distribution with mean λ and standard deviation $\sqrt{\lambda}$.

$$P(I) = \frac{\lambda^I}{I!} \exp\left[-\lambda\right] \approx \frac{1}{\sqrt{2\pi\lambda}} \exp\left[-\frac{(I-\lambda)^2}{2\lambda}\right] \tag{2}$$

In which case it can be difficult to distinguish between white and shot noise, although typically one process can dominate another depending on the image acquisition environment.

2.1.3 Salt-and-Pepper Noise

Salt-and-pepper noise is generated after image acquisition by errors occurring during data conversion or transmission. It is characterised by randomly distributed pixels displaying either the minimum or maximum intensity values allowed for the image data type. Therefore these pixels appear most prominently on dark backgrounds as white spots (salt) or on white backgrounds as dark spots (pepper). Consequently, salt-and-pepper noise can be mostly eliminated by using median filtering.

2.2 Preprocessing Steps

2.2.1 Gaussian Smoothing

In several steps we apply a Gaussian averaging algorithm, that smooths each pixel with the surrounding N pixels, weighted by the inter-pixel distance r using standard deviation of σ_G. For algorithmic simplicity, we refer to a 2D image in terms of a flattened 1D

vector $\mathbf{I}_i$. Therefore, the smoothing algorithm can be written as a convolution of pixel intensities $\mathbf{I}_i$ over the Gaussian kernel $\mathbf{G}_{ij}(\sigma_G)$, resulting in the smoothed image $\mathbf{S}_j$.

$$\mathbf{G}_{ij}(\sigma_G) = \frac{1}{\sqrt{2\pi}\sigma_G} \exp\left[-\frac{|j-i|^2}{2\sigma_G^2}\right] \tag{3}$$

$$\mathbf{S}_j = \sum^{N} \mathbf{I}_i \mathbf{G}_{ij}(\sigma_G) \tag{4}$$

2.2.2 Intensity Scaling

Acknowledging that any pixel intensities lying in the extreme regions of any image's distribution are likely to be artefacts due to shot or salt-and-pepper noise, we first rescale all pixel intensities to lie within the percentile range that would contain $p_0 \rightarrow p_1$ of all values. This is performed by clipping any extreme pixels with intensities lying outside the percentile range and then normalising by the resultant range of values.

$$\mathbf{I}_i = \frac{1}{I(p_1) - I(p_0)} \times \begin{cases} I(p_0) & if \quad \mathbf{I}_i < I(p_0) \\ I(p_1) & if \quad \mathbf{I}_i > I(p_1) \\ \mathbf{I}_i & else \end{cases} \tag{5}$$

This process is preferred to median filtering, since it does not result in a uniform loss in resolution. The default values of p_0 and p_1 can be found in the Appendix.

2.2.3 Non-Local Means Denoising

Most noise left in the resultant image is approximated to follow a Gaussian distribution, although now this may affect different regions to a varying extent. For example, empty areas will be dominated by white noise and so images containing little detail will appear

more noisy on the whole. In which case it is not appropriate to simply apply a global Gaussian filter across the entire image, as this would result in either insufficient noise removal in some areas or excess loss of resolution in others.

Therefore, we use the non-local means de-noising algorithm[2], which is designed to increase smoothing at regions in the image containing little detail and restrict smoothing at regions containing more detail or significant features.

Initially, for each pixel, designated by index i, the local mean value B is calculated from a local neighbourhood window of n pixels.

$$\mathbf{B}_i = \frac{1}{n} \sum^{n} \mathbf{I}_i \tag{6}$$

A smoothing factor $\mathbf{w}_{ij}$ is then calculated from the Gaussian-weighted Euclidean distance between each pairwise value of $\mathbf{B}_i - \mathbf{B}_j$, with an adjustable standard deviation σ_{NL}.

$$\mathbf{w}_{ij} = \exp\left[-\frac{|\mathbf{B}_i - \mathbf{B}_j|^2}{\sigma_{NL}^2}\right] \tag{7}$$

Note that the value of $\mathbf{w}_{ij}$ is invariant to the spatial proximity of pixels i and j, and only dependent on the local content at each position. Finally, the resultant de-noised image $\mathbf{D}_i$ is then generated by the convolution of the original image $\mathbf{I}_j$ and smoothing factor $\mathbf{w}_{ij}$ over a region of pixels specified by j. The convolution is also normalised by the integral of $\mathbf{w}_{ij}$ over the same region (equation (8)), which ideally would encompass the whole image. In practice, to perform this calculation would be computationally exhaustive, and so typically the integral region is limited to another local neighbourhood window of m pixels.

$$\mathbf{D}_i = \frac{\sum^{m} \mathbf{I}_j \mathbf{w}_{ij}}{\sum^{m} \mathbf{w}_{ij}} \tag{8}$$

Alternative implementations to the non-local means algorithm include replacing the Gaussian weighted smoothing factor with a term inversely proportional[3] to the Euclidean distance of $\mathbf{B}_i - \mathbf{B}_j$ (equation (9)) and performing the convolution in equation (8) in Fourier space[4].

$$\mathbf{w}_{ij} = \frac{1}{\sqrt{1 + |\mathbf{B}_i - \mathbf{B}_j|^2 + \sigma_{NL}^2}} \tag{9}$$

In practice, σ_{NL} is estimated from the distribution of white noise in the input image, and so is not a variable parameter. However, the neighbourhood window sizes n and m are variable, with default values listed in the Appendix.

2.2.4 Histogram Equalisation

An additional form of error present in images can come from systematic imbalance of intensities caused by unequal exposure of the biopsies to light. Therefore the distribution of pixel intensities across the image may appear higher in certain regions than others. A way to adjust for this effect is to apply a transformation at each pixel in order to enforce a linear cumulative distribution function of global pixel intensities, typically known as Histogram Equalisation (HE).

The definition of this transform varys between different HEs. For example, whereas an Ordinary Histogram Equalisation (OHE) uses a transformation derived from the intensity histogram for all pixels, an Adaptive Histogram Equalisation (AHE) uses a transformation derived from a local intensity histogram of some neighbourhood region for each pixel. We apply a Contrast Limiting Adaptive Histogram Equalisation (CLAHE)[5], which reduces the susceptibility of AHEs to increase noise in the image by enforcing a limit on the amplification of contrast.

3 Tensors

We use several filters and analysis metrics that utilise tensors derived from the first and second derivatives of each image pixel. Below is an overview of the terminology used and description of how each property is calculated.

3.1 Structure Tensor

The structure tensor can be used to estimate the local direction and magnitude of features in an image[6]. It is derived from the Jacobian matrix at each pixel, which can be averaged over a region if desired. Representing an image as a 2D intensity map $f(x, y)$, results in a Jacobian vector $\mathbf{J}(x, y)$ given in equation (10).

$$\mathbf{J}(x, y) = \left[\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y}\right] \tag{10}$$

The structure tensor $\mathbf{K}(x, y)$ is the dot product $\mathbf{J}(x, y) \cdot \mathbf{J}(x, y)^{\mathbf{T}}$. Representing the derivative of $f(x, y)$ with respects each variable as $f'_x(x, y)$ or $f'_y(x, y)$, we can represent the structural tensor as equation (11).

$$\mathbf{K}(x, y) = \mathbf{J}(x, y) \cdot \mathbf{J}(x, y)^{\mathbf{T}} = \begin{pmatrix} f'_x(x, y)^2 & f'_x(x, y) f'_y(x, y) \\ f'_y(x, y) f'_x(x, y) & f'_y(x, y)^2 \end{pmatrix} \tag{11}$$

In practice, the gradients $f'_x(x, y)$ and $f'_y(x, y)$ for a discrete data set such as $f(x, y)$ are often estimated by the central difference method or a spline based approach. The structure tensor of a region centred on $\mathbf{r}(x_0, y_0)$ can be estimated by averaging over a window (typically a Gaussian or uniform filter) $\mathbf{w}(\mathbf{r})$.

$$\mathbf{K}_w(x_0, y_0) = \int\int w(x - x_0, y - y_0)\, \mathbf{K}(x, y)\, dx\, dy \tag{12}$$

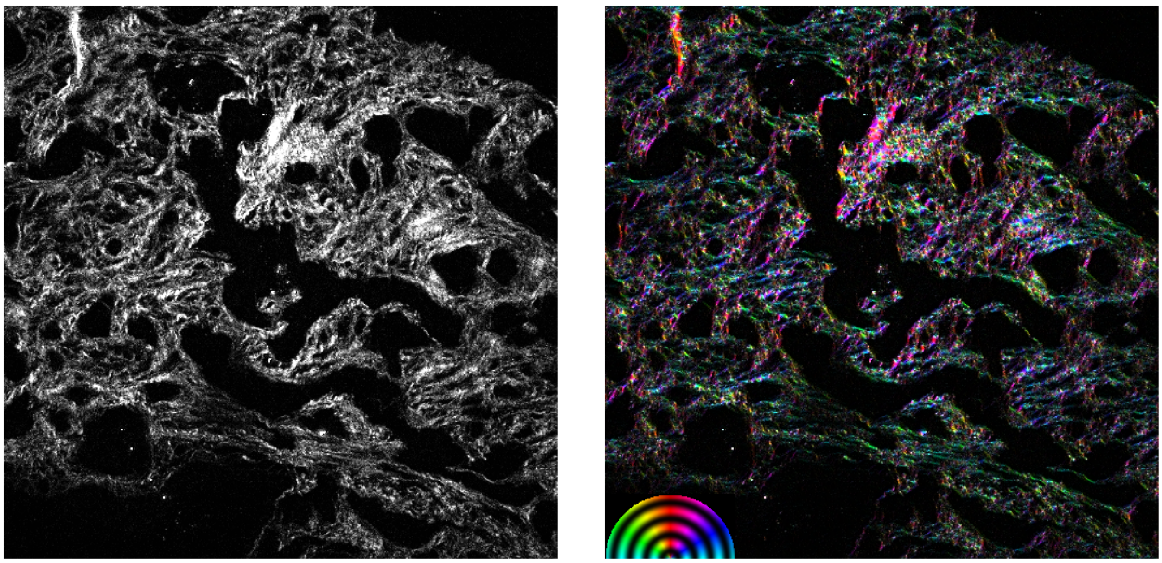


Figure 1: SHG greyscale (left) and structure tensor RGB (right) images, demonstrating fibril alignment

The maximum and minimum eigenvalues of the structure tensor, λ_{max}, λ_{min} and their corresponding eigenvectors $\mathbf{e}_{max}$, $\mathbf{e}_{min}$, can be used to measure the local anisotropy n, and orientation θ:

$$n = \lambda_{max} - \lambda_{min} \tag{13a}$$

$$\theta = \frac{1}{2} \arctan \left(\frac{2 \left\langle f'_x(x,y) f'_y(x,y,) \right\rangle}{\left\langle f'_y(x,y)^2 \right\rangle - \left\langle f'_x(x,y)^2 \right\rangle} \right) \tag{13b}$$

Using the structure tensor for each pixel allows us to form a RGB image representing the angular direction and magnitude of its content to demonstrate fibre alignment.

3.2 Nematic Tensor

Similar to the structure tensor, in order to measure the alignment of the collagen fibril network we adopt the same methodology as Garcia *et al.*[7], who employed the FibrilTool plugin[8] of the ImageJ software package[9]. This tool calculates the tangent to the structure unit vector $\mathbf{t}(x, y)$:

$$\mathbf{t}(x,y) = \left[\frac{-f'_y(x,y)}{\sqrt{f'_x(x,y)^2 + f'_y(x,y)^2}}, \frac{f'_x(x,y)}{\sqrt{f'_x(x,y)^2 + f'_y(x,y)^2}}, \right] \tag{14}$$

Which is transformed into components of the 2x2 nematic tensor $\mathbf{n}(x,y)$ for each pixel.

$$\mathbf{n}(x,y) = \mathbf{t}(x,y) \cdot \mathbf{t}(x,y)^{\mathbf{T}} = \begin{pmatrix} f'_y(x,y)^2 & -f'_x(x,y)f'_y(x,y) \\ -f'_y(x,y)f'_x(x,y) & f'_x(x,y)^2 \end{pmatrix} \frac{1}{f'_x(x,y)^2 + f'_y(x,y)^2} \tag{15}$$

The nematic tensor for a selected area then becomes the average local tensor $\langle \mathbf{n} \rangle$ of the constituent pixels. Once again, the eigenvalues and corresponding eigenvectors of $\langle \mathbf{n} \rangle$ are used to measure the anisotropy of collagen fibrils in the sampled image region.

3.3 Hessian Matrix

Similar to the Jacobian tensor, the Hessian matrix $\mathbf{H}(x,y)$ is derived from the second derivatives at position x, y (equation (16)).

$$\mathbf{H}(x,y) = \begin{pmatrix} f''_{xx}(x,y) & f''_{xy}(x,y) \\ f''_{yx}(x,y) & f''_{yy}(x,y) \end{pmatrix} \tag{16}$$

Consequently, the Hessian matrix is related to the Jacobian tensor by $\mathbf{H}(x,y) = \mathbf{J}(\nabla f(x,y))^{\mathbf{T}}$. It is also comparable with the local curvature, and therefore can provide more information on the boundaries of patterns within an image.

Similar to before, the Hessian matrix for a selected area becomes the average local tensor $\langle \mathbf{H} \rangle$ of the constituent pixels. The eigenvalues and corresponding eigenvectors of $\langle \mathbf{H} \rangle$ are used to measure the curvature of collagen fibrils in the sampled image region (see tubeness filter).

4 Filters

4.1 Hysteresis Threshold

Hysteresis thresholding applies an upper and lower intensity threshold (λ_l, λ_u) to an image to identify areas that are unlikely to be noise. Any area above the low threshold that is also connected to an area above the higher threshold can be considered as a continuation and therefore a robust feature.

Typically we set the lower threshold determined as either some fraction ($\approx 1/2$)of the mean intensity value, or the Li iterative minimum cross entropy method[10], whichever is higher.

The upper threshold is given by the ISODATA method[11], which satifies the equality given below.

$$\lambda_u = \bar{I}_{ISO}(\mathbf{I}, \lambda_u) + \frac{1}{2}\bar{I}_{ISO}(\mathbf{I}, \lambda_u) \tag{17}$$

$$\bar{I}_{ISO}(I, \lambda_u) = \frac{1}{N}\sum_{i}^{N} \begin{cases} \mathbf{I}_i & if \quad \mathbf{I}_i > \lambda_u \\ 0 & else \end{cases} \tag{18}$$

Another time we use hysteresis thresholding is to identify regions of overlap between two types of binary filter. This will be explored later in section 6.2.

4.2 Tubeness Filter

The tubeness filter $\mathbf{T}(x, y)$ is a relatively simple transform that uses the maximum eigenvalues H_{max} of the local Hessian matrix for each pixel (equation (19)).

$$\mathbf{T}(x,y) = \begin{cases} |H_{max}| & if \quad H_{max} < 0 \\ 0 & else \end{cases} \tag{19}$$

When combined with the FIbRe Extraction (FIRE) algorithm, it has been shown to effectively extract single fibres from SHG images of biopsies[12] .

4.3 Further Methods

The Hough transform is a feature extraction tool that was originally developed to identify lines and curves in images[13]. It was later extended for arbitrary shapes[14], and became a popular technique in the computer vision community. The Hough transform has been recently been implemented to measure the degradation of the cell cytoskeleton under ionising radiation[].

One of the most powerful and widely used tool to identify collagen fibres from medical images is CT-FIRE (Curvelet Transform–FIbeR Extraction software), found in the ImageJ[] suite. CT-FIRE uses the curvelet transform[15] (CT) (a wavelet transform variant) to filter out background image noise from individual fibres, making use of the CT's ability to resolve curved, cylindrical features. The software also contains a fibre extraction tool in order to quantify the structure of the collagen network[16]. [12]

5 Network Extraction

Fibril networks may be considered graphs, with the fibres themselves as edges between n nodes at the regions they interconnect. A graph can be described by an adjoint matrix, an nxn matrix that records whether an edge is present between each node in a system. The degree of each node determines the number of edges it possesses.

5.1 FIbRe Extraction (FIRE)

The FIRE (FIbRe Extraction) algorithm is designed to trace out a network on top of an image containing fibrous textures[16]. We use a modified version that is designed to extract fibrous detail at a higher resolution than previous versions.

The algorithm works by generating a network of nodes connected by edges that represent the outlines of fibres. A primary set of nucleation points is chosen, which become parent nodes that can then each propagate subsequent child nodes in the nearby region. The FIRE algorithm can therefore be modified by changing any rules required to generate the parent nodes and locate and connect subsequent child nodes.

Typically a filter can be applied to the input image to enhance any tubular-like regions. We use a simple "tubeness filter", based on the Hessian eigenvalues of each pixel, which has been shown[12] to have similar performance to the much more complex Curvelet transform (CT) method.

The FIRE algorithm starts by dividing the image into a binary representation of the foreground and background. We use the hysteresis threshold method, as explained in section 4.1 in order to determine which pixels should be included in the foreground. Then a distance matrix is computed representing the number of pixels lying between the foreground and background (see figure 2). We then apply a Gaussian filter in order to smooth the discrete distance matrix into continuous data, so as to reduce the number of redundant data values.

A number of nucleation points is generated at each local maximum on our smoothed distance matrix that lies above a threshold value (typically 2 pix). A set of parent nodes are then assigned to the pixel coordinates of these nucleation points. Propagation of subsequent child nodes is then performed by identifying further local maxima within a given search region of each parent node, termed Local Maximum Points (LMPs) (figure 2). If an LMP is a successful candidate then a new child node will be created at its

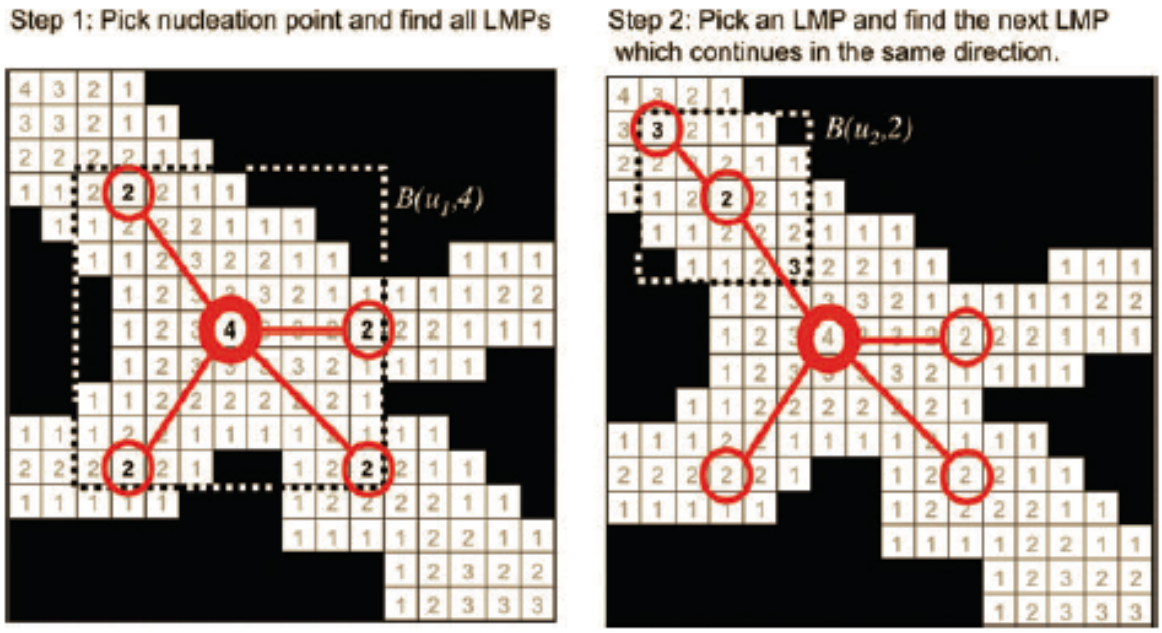


Figure 2: Figure taken from Stein 2008[16] representing propagation of LMP nodes from a nucleation node.

location, with an edge assigned to the parent node.

After the initial assignment of all parent and primary child nodes, propagation of the network is continued iteratively at each LMP until no further candidates of child nodes can be found. Typically a LMP will be accepted if it already has been assigned to an existing node or if it lies in the direction of propagation and possesses a distance matrix value above the threshold ϵ_0.

The pseudo code is shown below

for active child node i **do**

Search $\mathbf{D}$ for LMPs within window $B(\mathbf{r}_i)$

for local maxima candidate j **do**

if $\mathbf{D}_j \geq \epsilon_0$ **then**

Calculate angle θ_{ijn} between $\mathbf{r}_i$, $\mathbf{r}_j$ and parent node $\mathbf{r}_n$

end if

end for

Select j corresponding to minimum θ_{ijn}

if $\theta_{ijn} \leq \theta_0$ **then**

Calculate euclidean distance r_{ij} between at $\mathbf{r}_i$ and $\mathbf{r}_j$

if $r_{ij} \geq r_0$ **then**

Assign new child node j at $\mathbf{r}_j$ to parent n

Assign an edge between i and j

Activate node j

Deactivate node i

else

Move $\mathbf{r}_i$ to $\mathbf{r}_j$

end if

else

Calculate euclidean distance r_{ik} between at $\mathbf{r}_i$ and connected node $\mathbf{r}_k$

if $r_{ik} \geq r_0/10$ **then**

Remove edge between i and k

end if

Deactivate node i

end if

end for

Some cleaning is applied after all child nodes have been propagated in order to remove any artefacts from the FIRE algorithm.

1. If any 2 nodes lie within r_1 of each other, transfer edges from the node with the lowest degree to the node with the highest degree
2. Remove any nodes without edges
3. Remove any remaining connected networks that either contain only 1 node with 1 edge or only 1 node with 2 or more edges.

The differences between our implementation and the original FIRE algorithm includes the use of the edge length threshold r_0 and the ability to create an edge between any existing node (whereas originally only child nodes with the same parent node could be connected). The use of r_0 allows us to both move the LMPs after their original assignment and also control the resolution at which each fibre can be traced without altering the size of the local LMP search window B.

5.2 Network Manipulation

We can manipulate the raw network $\mathbf{R}$ in two main ways: either to provide information about individual fibres or to give us an insight into its overall structure and connectivity. In order to do either, however, some extra processing is required.

5.2.1 Assigning Fibres

In order to calculate properties such as fibre length and waviness we need to be able to identify individual fibres from within each network. This is not trivial and depends solely on the rules assigned to define what an appropriate fibre is. However, we apply a very similar approach to the FIRE algorithm in order to do so.

To begin with we identify all "external" nodes that only contain 1 edge and so therefore will definitely reside at the start or and of any fibre. These then become our parent nodes for a fibre, and their primary child node will be the single node that they share an edge with. The algorithm then traces back along any nodes connected to the active child node and rebuilds a fibre based on a set of rules.

for active child node i **do**

  for each connected node j **do**

    Calculate angle θ_{ijn} between $\mathbf{r}_i$, $\mathbf{r}_j$ and parent node $\mathbf{r}_n$

  end for

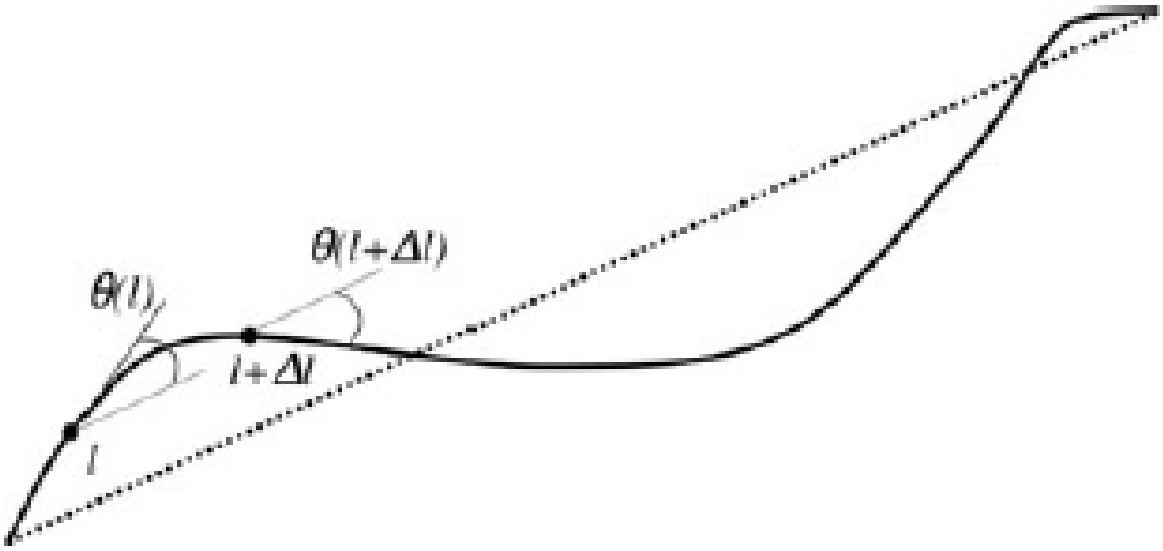


Figure 3: Figure taken from Stein 2008[16] representing measurement of a single fibre 'waviness' property.

Select j corresponding to minimum θ_{ijn}

if $\theta_{ijn} \leq \theta_0$ **then**

Assign new child node j at $\mathbf{r}_j$ to parent n

Activate node j

end if

Deactivate node i

end for

5.2.2 Network Reduction

We can apply graph theory to investigate the structure of collagen networks, but

6 Segmentation

6.1 Network Segmentation

Once the global fibre network has been generated using FIRE, we can segment the image based on regions of fully connected sub-networks. It its not guaranteed that the global network will be fully connected (i.e. a path can be drawn between any 2 nodes), since each nucleation point is propagated independently. Therefore it becomes possible to treat each connected sub-network separately.

The segmentation process begins by creating a binary matrix containing straight lines between each network node and performing a binary dilation for a set number of iterations. A Gaussian filter is then applied in order to smooth the fibre edges.

6.2 Colour Segmentation

An alternative method of segmentation uses clustering of RGB values in a colour image. This is commonly used for microscope-stained HE images in software packages such as CurveAlign[], since typical gram-straining yields cellular regions of a standard pigment. However, both SHG and PL imaging techniques produce greyscale images, so a way to combine them into a composite RGB data format must be developed first.

6.2.1 Composite RGB Image

We create a composite 3 channel image from the SHG, PL and PL transmission data, corresponding to red, blue and green channels respectively, taken at a fixed biopsy region.

Each RGB pixel vector is then normalised into a unit vector, in order to reduce the dependence on lighting intensity of each image type. Therefore we expect fiborous features to show up as unit vectors with large red components, whereas cellular features should

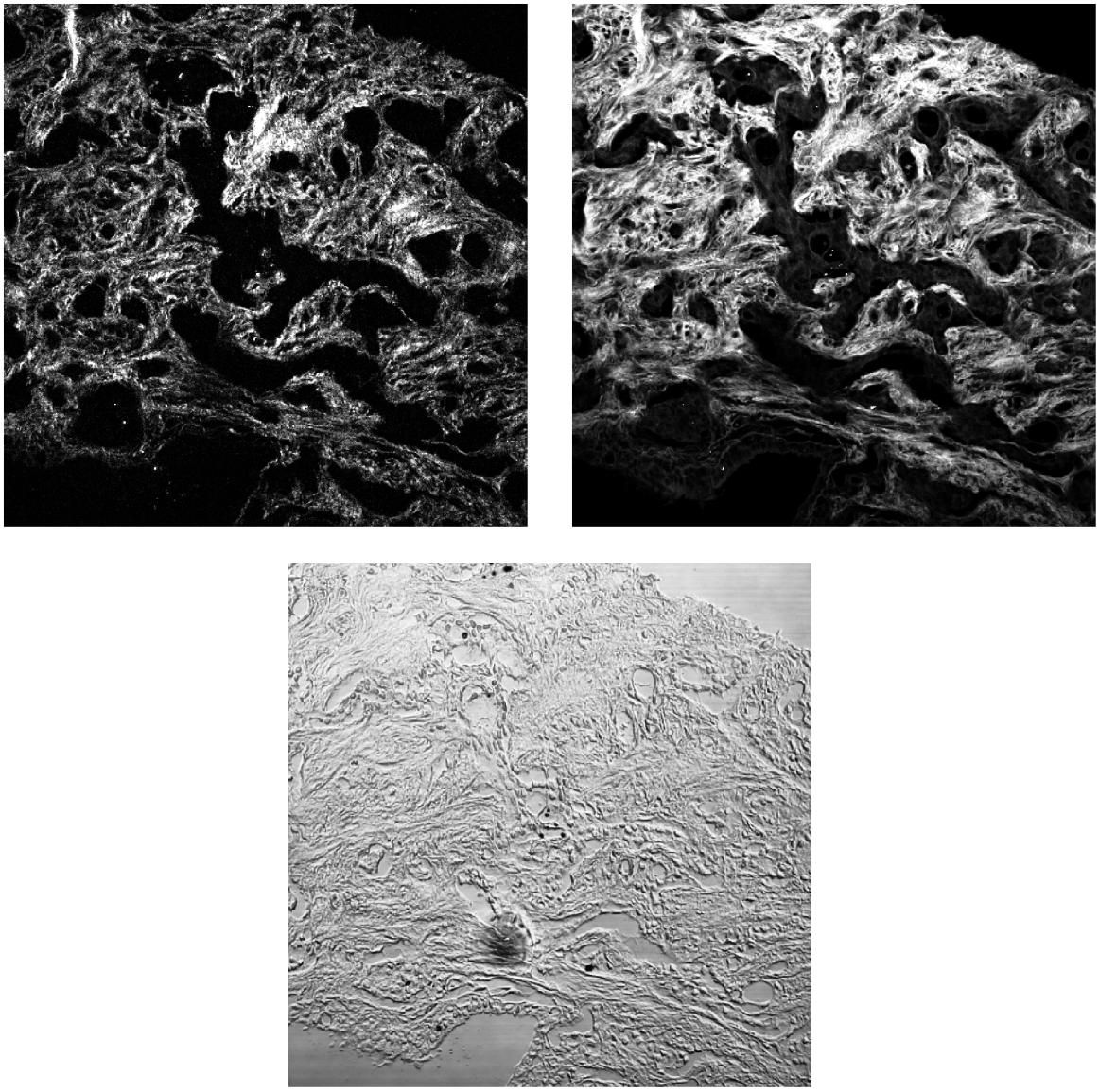


Figure 4: SHG greyscale (left) and structure tensor RGB (right) images, demonstrating fibril alignment

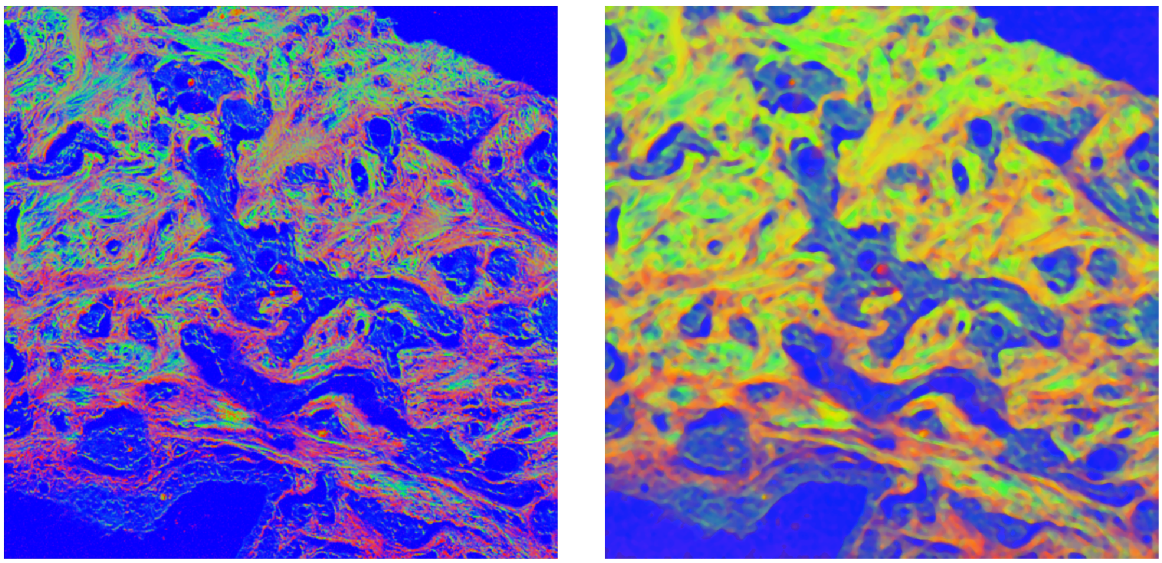


Figure 5: SHG (top left), PL (top right) and PL transmission (bottom) greyscale images

result in unit vectors containing prominently large green and blue components.

In order to enhance the clustering of colours and yield smooth segments we apply some smoothing techniques on the raw composite image. These techniques mimic those carried out by the CurveAlign software BDcreationHE algorithm. To begin with we apply a contrast stretching routine to each channel in order to maximise their differences. The channels are then padded around the outside with a region of zeroed pixels and a intensity histogram equalisation algorithm is applied, as detailed in section 2.2.4. Finally, a median filter is used to smooth the image, before the extra padded pixels are removed. The effect of this procedure is demonstrated in figure 5.

6.2.2 Kmeans Clustering

Segmentation of the image is performed by clustering of each pixel's RGB components, independent of its location. Considering that a typical biopsy image contains $512 \times 512 = 262144$ pixels and therefore $512 \times 512 \times 3 = 786432$ data points, a batch implementation of any clustering algorithm needs to be used for computational efficiency. We use the MiniBatchKMeans function as implemented in scikit learn, which has been shown to

achieve very similar performace to its full KMeans implementation[]. It must be noted that KMeans is a stochastic algorithm, and therefore is not guaranteed to return the optimum solution at every run.

After clustering of the main 8 colours present in our RGB composite image, we then proceed to assign whether these clusters should be considered either cellular or fibrous, based on their centroid unit vector $\mathbf{c}$ and average non-zero intensity $\bar{I}$ values. We consider the cluster to contain cellular features if it contains a non-zero value of $\phi(\mathbf{c}, \bar{I})$. The terms in $\phi(\mathbf{c}, \bar{I})$ consist of 3 angles on the RGB colour sphere corresponding to the unit vector $\mathbf{c}$ as well as $\bar{I}$. It should be noted that the numerical limits in equation (20) are of a heuristic nature and therefore serve only as a rough colour guide, since we cannot rely fully on the reproducibility of RGB distributions between our composite images.

$$\phi(\mathbf{c}, \bar{I}) = \prod \begin{cases} 1 & if \quad \arcsin(\mathbf{c}_R) < 0.7 & else \quad 0 \\ 1 & if \quad \arcsin(\mathbf{c}_G) < 1.1 & else \quad 0 \\ 1 & if \quad \arccos(\mathbf{c}_B) < 1.4 & else \quad 0 \\ 1 & if \quad \bar{I} < 0.8 & else \quad 0 \end{cases} \tag{20}$$

The pixels present in these accepted clusters are then combined to form our binary "cell filter", which we use as the basis for further image segmentation. In order to deal with the stochastic nature of the KMeans algorithm we rank each cell cluster by the summation of each term in equation (20), resulting in the cost function Ψ (equation (21)). The KMeans run that creates the lowest average Ψ is then chosen as the optimal solution for our cell filter.

$$\Psi = \frac{1}{N} \sum_{i}^{N} \arcsin(\mathbf{c}_{R_i}) + \arcsin(\mathbf{c}_{G_i}) + \arccos(\mathbf{c}_{B_i}) + \bar{I}_i \tag{21}$$

Finally, we deal with any segments that may be artefacts by enforcing a minimum pixel area.

7 Analysis Methods

A variety of image processing techniques have been used to assess the orientation and structure of collagen fibres in the ECM. Most have been applied using the hypothesis that there is a relationship between cancer progression and the alignment of fibres. To this end, the techniques also attempt to define a metric to describe this alignment. These include Fourier[17] and nematic tensor[8] analysis of image intensity as well as more advanced methods to identify individual fibrils employing the curvelet transform[12]. However, generally there is no commonly agreed automated method to measure cancer progression from medical imaging. We outline a range of image analysis techniques below that have either been used to cancer biopsies, or that may prove beneficial for future investigations.

7.1 Fourier Transform

Fourier analysis methods have been used previously to quantify the orientation distribution $P(\phi)$ of fibres in a SHG image[17]. They can be quick and easy to implement using fast Fourier transform (FFT) methods, such as those available in NumPy. The Fourier transform A_{jk} of our 2D SHG images then becomes

$$A_{jk} = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} I_{uv} \exp\left(2\pi i \left(\frac{uj}{N} + \frac{vk}{M}\right)\right) \tag{22}$$

With the wave amplitudes α_{kj} and angles ϕ_{kj} for each frequency corresponds to equation (23).

$$\alpha_{jk} = \sqrt{A^*_{kj} A_{kj}} \tag{23a}$$

$$\phi_{jk} = \text{Imag}\left\{\ln\left(\frac{A_{kj}}{\alpha_j k}\right)\right\} \tag{23b}$$

We calculate the probability distribution $P(\phi)$ of fibril orientations using the average amplitudes $\langle\alpha(\phi_j k)\rangle$

$$P(\phi) = \langle\alpha(\phi_{jk})\rangle \quad where \quad \phi_{jk} = \phi \tag{24}$$

The spectrum of $P(\phi)$ is then able to inform us of the likelihood of finding fibres in a particular orientation. Although the value of ϕ is arbitrary, the distribution of probabilities can inform us of how ordered the system is. For example, a highly ordered system is likely to produce a spectrum dominated by a few values of $P(\phi)$, whereas a relatively disordered system will have a uniform distribution of $P(\phi)$.

Metrics used to describe the Fourier spectrum include the spectral directional index (SDI) and modified directional index (MDI), which have been recently applied to AFM images in order to measure the degradation of cell cytoskeleton fibres under ionising radiation[18].

8 Parameter Defaults

Table **??** list parameter defaults used in PyFibre. These can be changed by selecting the parameter section.

Parameter	Default Value
$sigma_G$ (pix)	0.5
p_0 (%)	1
p_1 (%)	98
n (pix)	12×12
m (pix)	35×35

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Prentic Hill, Upper Saddle River, New Jersey, USA, 2002.

[2] A. Buades, B. Coll, and J. . Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 60–65, June 2005.

[3] J. Darbon, A. Cunha, T. F. Chan, S. Osher, and G. J. Jensen, "Fast nonlocal filtering applied to electron cryomicroscopy," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1331–1334, May 2008.

[4] J. Froment, "Parameter-free fast pixelwise non-local means denoising," *Image Processing On Line*, vol. 4, p. 300–326, 2014.

[5] P. Heckbert, *Graphics Gems IV*.

[6] B. J ahne, *Digital Image Processing*. Springer-Verlag Berlin Heidelberg, 2005.

[7] A. M. Garcia, F. L. Magalhes, J. S. Soares, E. Paulino-Jr, M. F. de Lima, M. Mamede, and A. M. de Paula, "Second harmonic generation imaging of the collagen architecture in prostate cancer tissue," *Biomed. Phys. Eng. Express*, vol. 4, p. 025026, 2018.

[8] A. Boudaoud, A. Burian, D. Borowska-Wykret, M. Uyttewaal, R. Wrzalik, D. Kwiatkowska, and O. Hamant, "Fibriltool, an imagej plug-in to quantify fibrillar structures in raw microscopy images," *Nat. Protocols*, vol. 9, p. 457–463, 2014.

[9] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, "Nihimage to imagej: 25 years of image analysis," *Nat. Methods*, vol. 9, p. 671–675, 2012.

[10] C. Li and P. Tam, "An iterative algorithm for minimum cross entropy thresholding," *Pattern Recognition Letters*, vol. 19, no. 8, pp. 771 – 776, 1998.

[11] T. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, pp. 630–632, Aug 1978.

[12] J. S. Bredfeldt, Y. Liu, C. A. Pehlke, M. W. Conklin, J. M. Szulczewski, D. R. Inman, P. J. Keely, R. D. Nowak, T. R.Mackie, and K. W. Eliceiri, "Computational segmentation of collagen fibers from second-harmonic generation images of breast cancer," *J. Biomed. Opt.*, vol. 19, p. 16007, 2014.

[13] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Comm. ACM.*, vol. 15, pp. 11–15, 1972.

[14] D. H. Ballard, "Generalising the hough transform ro detect arbitrary shape," *Pattern Recognition*, vol. 13, pp. 111–122, 1981.

[15] J. L. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans Image Process*, vol. 11, p. 670–684, 2002.

[16] A. M. Stein, D. A. Vader, L. M. Jawerth, D. A. Weitz, and L. M. Sander, "An algorithm for extracting the network geometry of three- dimensional collagen gels.," *J. Microsc.*, vol. 232, p. 463–475, 2008.

[17] A. Ghazaryan, H. F. Tsai, G. Hayrapetyan, W.-L. Chen, Y.-F. Chen, M. Y. Jeong, C.-S. Kim, C. S-J., and D. C-Y., "Analysis of collagen fiber domain organization by fourier second harmonic generation microscopy," *J. Biomed. Opt.*, vol. 18, p. 031105, 2012.

[18] M. Manghi, L. Bruni, and S. Croci, "MDI: Integrity Index of Cytoskeletal Fibers Observed by AFM," *The European Physical Journal Plus*, vol. 131, no. 6, p. 213, 2016.