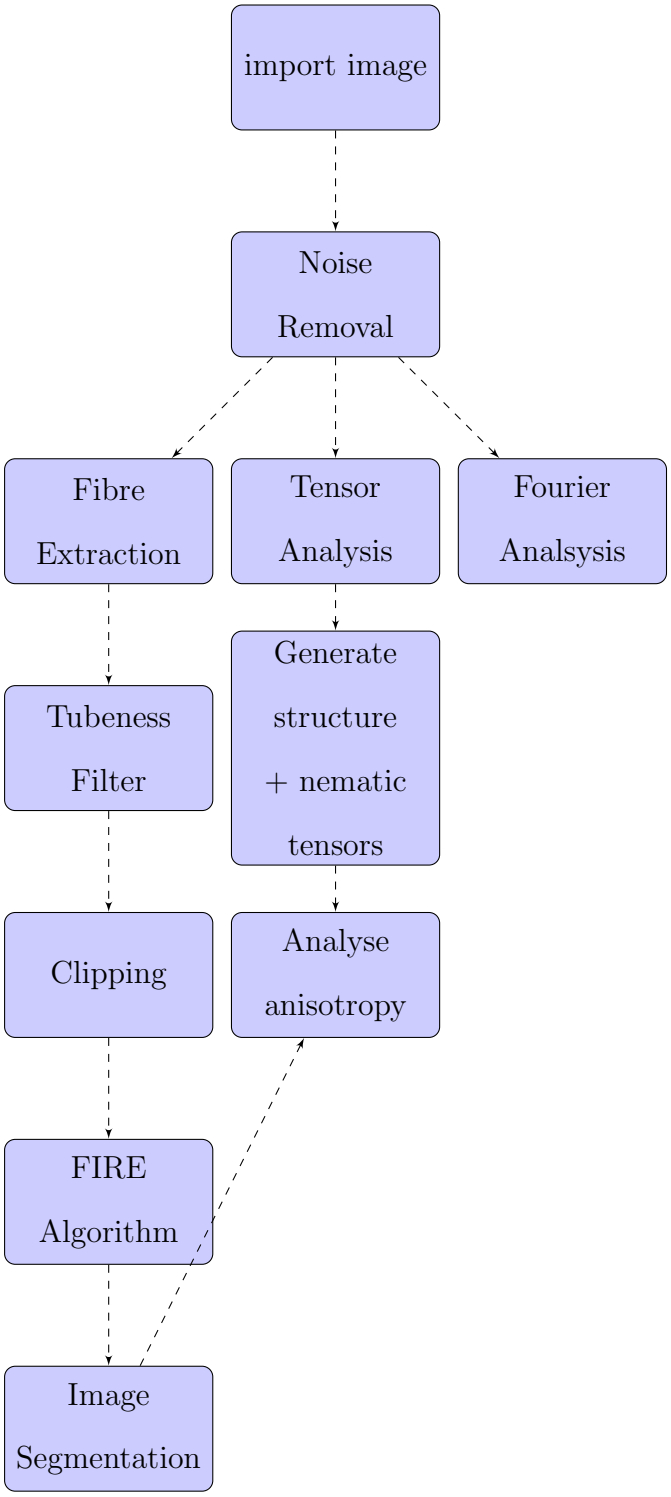
PyFibre - Python Fibrous Image Analysis Toolkit

Francis G.J. Longford

January 22, 2019

PyFibre (Python Fibrous Image Analysis Toolkit) is a computer program written in Python purposely built to analyse fibrous tissue.

1 Flowchart



1. Preprocessing of input images, including high and low-pass filtering followed by adaptive equalisation and Gaussian convolution.

2. Form nematic tensor using Gaussian convolution with a standard deviation of $\sigma = 0.5$ px
3. Form structure tensor using Gaussian convolution with a standard deviation of $\sigma = 0.5$ px
4. Form Hessian matrix using Gaussian convolution with a standard deviation of $\sigma = 0.5$ px
5. Calculate "tubeness" T at each pixel
6. Perform a modified version of the FIRE algorithm to extract

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 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. For algorithmic simplicity, we refer to a 2D image in terms of a flattened 1D vector $\mathbf{I}_i$.

$$\mathbf{I}_i = \frac{1}{I(p_1) - I(p_0)} \times \begin{cases} I(p_0) & if \quad \mathbf{I}_i < I(p_0) \\ I(p_1) & if \quad \mathbf{I}_i > I(p_1) \\ \mathbf{I}_i & else \end{cases} \tag{3}$$

PyFibre used default values of $p_0 = 1\%$ and $p_1 = 98\%$. This process is preferred to median filtering, since it does not result in a uniform loss in resolution.

2.2.2 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.

$$\mathbf{B}_i = \frac{1}{n}\sum^{n}\mathbf{I}_i \tag{4}$$

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.

$$\mathbf{w}_{ij} = \exp\left[-\frac{|\mathbf{B}_i - \mathbf{B}_j|^2}{\sigma^2}\right] \tag{5}$$

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 (6)), 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.

$$\mathbf{D}_i = \frac{\sum^{m}\mathbf{I}_j\mathbf{w}_{ij}}{\sum^{m}\mathbf{w}_{ij}} \tag{6}$$

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 (7)) and performing the convolution in equation

(6) in Fourier space[4].

$$\mathbf{w}_{ij} = \frac{1}{\sqrt{1 + |\mathbf{B}_i - \mathbf{B}_j|^2 + \sigma^2}} \tag{7}$$

Typical values of $n = 7 \times 7$ grid and $m = 35 \times 35$ grid.

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[5]. 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 (8).

$$\mathbf{J}(x, y) = \left[\frac{\partial f(x, y)}{\partial x}, \ \frac{\partial f(x, y)}{\partial y} \right] \tag{8}$$

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 (9).

$$\mathbf{K}(x, y) = \mathbf{J}(x, y) \cdot \mathbf{J}(x, y)^{\mathbf{T}} = \begin{pmatrix} f'_x(x, y)^2 & f'_x(x, y) f'_y(x, y) \\ f'_y(x, y) f'_x(x, y) & f'_y(x, y)^2 \end{pmatrix} \tag{9}$$

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) $\mathbf{w}(\mathbf{r})$.

$$\mathbf{K}_w(x_0,y_0) = \int\int w(x-x_0, y-y_0)\,\mathbf{K}(x,y)\,dx\,dy \tag{10}$$

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:

$$n = \lambda_{max} - \lambda_{min} \tag{11a}$$

$$\theta = \frac{1}{2}\arctan\left(\frac{2\left\langle f'_x(x,y) f'_y(x,y,)\right\rangle}{\left\langle f'_y(x,y)^2\right\rangle - \left\langle f'_x(x,y)^2\right\rangle}\right) \tag{11b}$$

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.*[6], who employed the FibrilTool plugin[7] of the ImageJ software package[8]. This tool calculates the tangent to the structure unit vector $\mathbf{t}(x,y)$:

$$\mathbf{t}(x,y) = \left[\frac{-f'_y(x,y)}{\sqrt{f'_x(x,y)^2 + f'_y(x,y)^2}}, \frac{f'_x(x,y)}{\sqrt{f'_x(x,y)^2 + f'_y(x,y)^2}},\right] \tag{12}$$

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

$$\mathbf{n}(x,y) = \mathbf{t}(x,y) \cdot \mathbf{t}(x,y)^{\mathbf{T}} = \begin{pmatrix} f'_y(x,y)^2 & -f'_x(x,y)f'_y(x,y) \\ -f'_y(x,y)f'_x(x,y) & f'_x(x,y)^2 \end{pmatrix} \frac{1}{f'_x(x,y)^2 + f'_y(x,y)^2} \tag{13}$$

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 (14)).

$$\mathbf{H}(x,y) = \begin{pmatrix} f''_{xx}(x,y) & f''_{xy}(x,y) \\ f''_{yx}(x,y) & f''_{yy}(x,y) \end{pmatrix} \tag{14}$$

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 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 (15)).

$$\mathbf{T}(x,y) = \begin{cases} |H_{max}| & if \quad H_{max} < 0 \\ 0 & else \end{cases} \tag{15}$$

4.2 Curvelet Transform

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[9] (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[10]. [11]

4.3 Hough Transform

The Hough transform is a feature extraction tool that was originally developed to identify lines and curves in images[12]. It was later extended for arbitrary shapes[13], 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[]. Considering the similarities between the appearance of a stiff collagen fibre network and ...., it is likely that the same technique could also be applied to measure the anisotropy of the ECM.

5 Network Extraction

5.1 Network Theory

Fibril networks may be considered graphs, with the fibres themselves as edges between n nodes at the regions they interconnect. A graph cant be described by an adjoint matrix, an nxn matrix that records whether an edge is present between each node in a system. represents a collection of nodes Before conversion to a graph

5.1.1 FIbRe Extraction (FIRE)

The FIRE (FIbRe Extraction) algorithm is designed to trace out a network on top of an image containing fibrous textures.

5.2 Hierarchical Clustering

Until this point we have considered image analysis techniques that attempt to define a metric based on the rationale of fibre alignment. However, it is also possible to measure the similarity of images without the need for quantification of some known or commonly agreed features. This approach may be used to investigate the appropriateness of the classification of biopsies based on features outlined in the Gleason score (figure **??**). Simply put, it is beneficial to assess whether a machine would classify a set of images in the same way that a trained human would.

6 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[14] and nematic tensor[7] analysis of image intensity as well as more advanced methods to identify individual fibrils employing the curvelet transform[11]. 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.

6.1 Fourier Transform

Fourier analysis methods have been used previously to quantify the orientation distribution $P(\phi)$ of fibres in a SHG image[14]. 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

$$A_{jk} = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} I_{uv} \exp\left(2\pi i \left(\frac{uj}{N} + \frac{vk}{M}\right)\right) \tag{16}$$

With the wave amplitudes α_{kj} and angles ϕ_{kj} for each frequency corresponds to equation (17).

$$\alpha_{jk} = \sqrt{A^*_{kj} A_{kj}} \tag{17a}$$

$$\phi_{jk} = \text{Imag}\left\{\ln\left(\frac{A_{kj}}{\alpha_j k}\right)\right\} \tag{17b}$$

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

$$P(\phi) = \langle\alpha(\phi_{jk})\rangle \quad where \quad \phi_{jk} = \phi \tag{18}$$

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 probabil-

ities 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[15].

6.2 Advanced Methods

6.2.1 Non-Negative Matrix Factorisation (NMF)

We consider a matrix solely comprised of non-negative elements $\mathbf{V}$ to be able to be factorised into two smaller matrices $\mathbf{W}$ and $\mathbf{H}$.

$$\mathbf{V} = \mathbf{WH} \tag{19}$$

The features matrix $\mathbf{W}$, represents a set of key (hidden) features, and the coefficients matrix $\mathbf{H}$, provides the weighting of these features in the original $\mathbf{V}$. The factorisation relies on the property that all matrices do not possess any negative elements, making it particularly applicable to image analysis. Solutions to the matrices $\mathbf{H}$ and $\mathbf{W}$ are estimated by the following criteria, where $||A||_F$ signifies the Frobenius norm (equation (21)).

$$\min_{W,H} ||V - WH|\,|_F^2 \qquad \text{where} \qquad W \geq 0, H \geq 0 \tag{20}$$

$$||A||_F^2 = \sum_{i,j} A_{ij}^2 \tag{21}$$

In practice, there are several algorithms available to find a solution to equation (20). One of the most popular and straight forward methods is the multiplicative update rule, developed by Lee and Seung[16]. Using an initial (non-negative) guess of $\mathbf{W}$ and $\mathbf{H}$, each index i, j is iteratively updated at step n using the following scheme, until both matrices are stable.

$$H_{ij}^{n+1} \leftarrow \mathbf{H}_{ij}^{n} \frac{((W^n)^T V)_{ij}}{((W^n)^T W^n H^n)_{ij}} \tag{22}$$

$$W_{ij}^{n+1} \leftarrow \mathbf{W}_{ij}^{n} \frac{(V(H^{n+1})^T)_{ij}}{((W^{n+1} H^{n+1} (H^{n+1})_{ij}^T} \tag{23}$$

6.2.2 Neural Networks

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *In NIPS*, pp. 556–562, MIT Press, 2000.