**CABANA (C**oll**A**gen Fi**b**re **ANA**lyzer)

This program was developed for analysis of collagen fibre architecture in IHC and fluorescence images.

Cabana can be run on a desktop PC or on Google Cloud.

For questions, please contact Xufeng Lin ([x.lin@garvan.org.au](mailto:x.lin@garvan.org.au)) or Astrid Magenau ([a.magenau@garvan.org.au](mailto:a.magenau@garvan.org.au)). Cabana on Google Cloud has been implemented by Tim Kallady ([t.kallady@garvan.org.au](mailto:t.kallady@garvan.org.au)).

**Hardware requirements for desktop version**

1. **Processor (CPU):** Intel Core i7 or AMD Ryzen 7, or higher for better performance
2. **Memory (RAM):** Minimum 16 GB, with 32 GB preferred
3. **Graphics Processing Unit (GPU):** NVIDIA GeForce GTX/RTX series with at least 10 GB VRAM

**Cabana installation**

Please note that Cabana can only run efficiently on computers that full-fill the minimum requirements above.

1. The source code of Cabana is hosted at [Github](https://github.com/Garvan-Data-Science-Platform/Cabana). Download, unzip and place the “Cabana” folder in “C:\Program Files\”
2. Install Anaconda or Miniconda
3. When running the program for the first time, launch "Anaconda Prompt":
   1. move the mouse to the Windows Icon (lower left), type "Anaconda" and select "Anaconda Prompt"
   2. a command window named "Anaconda Prompt" will open
   3. type “conda init” in the command window. Press enter.
4. Copy the "TWOMBLI" folder located in "C:\Program Files\Cabana\" to your personal user directory, such as "D:\your\_user\_name\project", so that you will have your own copy of TWOMBLI and associated parameter files, which you can modify freely.
5. There are 3 folders inside the “TWOMBLI” folder:

* ‘Programs’ containing Twombli ImageJ macro (Twombli\_v5.ijm is the latest version), Anamorf properties (AnamorfProperties.xml), Segmenter Parameters (SegmenterParameters.txt) and Twombli parameters (TwombliParameters.txt).
* 'Input': file location for images to be processed.
* 'Output': file location for processed files and results.

1. Copy the "Cabana" shortcut located in "C:\Program Files\Cabana" to your desktop.
2. Cabana relies on specific versions of ImageJ plugins. Copy the two jar files (anamorf-3.0.3.jar and ij\_ridge\_detect-1.4.2-SNAPSHOT.jar) in “C:\Program Files\Cabana\TWOMBLI" to “C:\Program Files\fiji-win64\plugins\”. Overwrite existing jar files if needed.

**How to run the desktop version of Cabana**

1. Launch the program by double-clicking the Cabana icon on desktop.
2. Once launched, the program will sequentially ask for the locations of the 'Programs', 'Input', and 'Output' folders.
3. Select the respective folder location.
4. Press ok and the analysis will commence.

**How to run Cabana on Google Cloud**

Cabana can be run on Google Cloud. This is recommended for data sets in excess of 100 images and significantly reduces the processing time compared to the Desktop installation. Please note that this is only available for Garvan staff and students at the moment.

1. Add all images, SegmenterParameters.txt and TwombliParameters.txt to a .zip file.

Note: optimise the Segmenter and Twombli parameters on the Desktop version (see below) before running the entire data set on Google Cloud.

1. Go to [cabana.dsp.garvan.org.au](http://cabana.dsp.garvan.org.au/) and login with your Garvan credentials
2. Name your project in ‘Project name’.
3. Click ‘Create Project’.
4. Click ‘Choose File’ to upload the .zip file containing the data set and the two parameter files. Note: The parameter files must be included in the .zip file.
5. Once processing has been completed, the project will appear in the ‘Results’ tab.
6. Click on the links to download results.
7. Unzip results with 7-zip.
8. Delete results from Cabana dashboard at [cabana.dsp.garvan.org.au](http://cabana.dsp.garvan.org.au/)

**Parameter Selection**

**Segmenter Parameters**

This component of Cabana aims to extract collagen fibre areas determined by Picrosirius Red staining in an image from cluttered background based on colour and other low-level features. It relies on a self-supervised semantic segmentation model based on convolutional neural networks to group semantically similar neighbouring pixels. The mean colour of the pixels in the same segment will be compared with a user-specified threshold to determine whether the segment is the region of interest (ROI).

Note: The semantic segmentation algorithm is intended to only work for Picrosirius Red images. For extracting ROI of other colours, you will need to adjust the Normalized Hue Value in ‘SegmenterParameter.txt’ accordingly**. The segmentation model does not work well for grayscale images.**

The following image segmentation parameters for ROI extraction can be specified in ‘SegmenterParameter.txt’:

1. **Number of Labels**

The number of labels for semantic segmentation. It controls the level of granularity of segmentation. A higher number of labels will result in segments of smaller size. The default value is 48.

1. **Max Iterations**

The maximal number of iterations for semantic segmentation. This value should be large enough to generate reliable segmentation results but not too high to avoid grouping all pixels into a single segment. The default value is 30.

1. **Normalized Hue Value**

The normalized hue value in [0,1] for the colour of interest in HSB/HSV colour space. The typical hue values for green, blue, and red colour are 0.33, 0.66, and 1.0, respectively.

1. **Color Threshold**

Colour threshold used to determine ROI. Only segments with a mean colour greater than this threshold will be extracted as ROI. The default value is 0.2.

1. **Min Size**

The minimal size of segments. Any segment with a size smaller than this parameter will be ignored. The default value is 64.

1. **Max Size**

The maximum allowable image size. Any image with a size larger than the \*square\* of this parameter will be ignored by the program. The default value is 2048.

1. **Mode**

Three modes are supported:

1. "segmentation": only image segmentation is performed. Use to test segmentation parameters.
2. "twombli": image analysis is applied to the original image (not the ROIs output by the segmentation). Select ‘Twombli’ if analysing fluorescent images, i.e., light fibres on dark background where image segmentation is not necessary.
3. "both": image analysis is applied to the region of interest, i.e., both segmentation and Twombli will be executed.

Note: When the boundaries between regions of interest (Picrosirius Red areas) and backgrounds are not clearly distinguishable, obtaining satisfactory segmentation results can be difficult. To address this, the "Number of Labels" and "Max Iterations" parameter can be optimised. Increasing the number of labels generally leads to finer granularity in segmentations, which may result in a better segmentation result. Avoid increasing the number of labels beyond 64, otherwise GPU Memory might overflow. On the other hand, decreasing the max iterations may result in premature segmentation, reducing the likelihood of mixing fibres and background. Adjusting the HUE parameter may not have much of an effect as the colour of Picrosirius stained collagen fibres is already red/pink.

**Collagen fibre analysis**

This component detects and quantifies fibre structures in images using Fiji plugins including Ridge Detection (<https://imagej.net/plugins/ridge-detection>), Anamorf (<https://github.com/djpbarry/AnaMorf/wiki>). Optional Gap Analysis was also implemented in Python using max inscribed circles (<https://imagej.net/plugins/max-inscribed-circles>). The parameters of Anamorf can be specified in 'AnamorfProperties.xml', while those of Ridge Detection and the optional gap analysis can be specified in 'TwombliParameters.txt'.

Please refer to this document <https://github.com/wershofe/TWOMBLI/blob/master/TWOMBLI_v1/TwombliDocumentation.docx> for more details about parameter settings.

The following parameters must be specified in ‘TwombliParameters.txt’:

1. **Dark Line**

Set to 0 = the program assumes that fibres are light on a dark background (fluorescence, birefringence, SHG etc).

Set to 1 = Dark fibres on a light background (Picrosirius Red IHC).

1. **Contrast Saturation**

A value between [0,1] shows the percent of pixels that will be saturated for contrast enhancement.

1. **Min Line Width**

It defines the minimum line (ridge) width in pixels that the ridge detection algorithm can detect. The line width is used to estimate the 'Sigma' parameter of Gaussian filtering kernel: .

1. **Max Line Width**

The maximum line (ridge) width in pixels that the ridge detection algorithm can detect.

Twombli runs ridge detection repeatedly with every value between the minimum and maximum line width and calculates the combination of the detected ridges.

Note:

Setting a large line width (e.g., >15) gives rise to the chance of 'straight line artefacts', which are caused by the small (close to zero) upper threshold of filtering response.

1. **Line Width Step**

This parameter controls the sampling factor for line widths between the minimum and the maximum line width. It allows for increasing the line width by step larger than 1.

For instance, if you want to detect ridges with multiple line widths 5, 7, and 9, you can specify a line width step to 2 with the min line width and the max line width . The default value is 1, which means all line widths between the min and max line width will be used for the ridge detection.

1. **Low Contrast**

Defines the lowest grayscale contrast between a line (collagen fibre) and background (non-fibre area).

This parameter is used to estimate the lower threshold for the filtering response: Line points with a filtering response lower than the low contrast threshold are discarded.

1. **High Contrast**

The highest grayscale contrast between a line (collagen fibre) and background (non-fibre area). This parameter is used to estimate the upper threshold for filtering response: . Line points with a response larger than the high contrast threshold are accepted. Line points with a response in [ are added to the accepted line points if line structures are reasonably formed.

In other words: not the absolute intensity is important, but the difference/contrast of a pixel with its neighbours. This applies to low and high contrast.

If a pixel with an intensity of 230 appears among neighbouring pixels with intensity of 230, the filtering response will be zero because this point is not visually salient. If this pixel appears among pixels with an intensity of 10, then the filtering response will be strong enough to signify it as a salient line/ridge point. For example, pixels values > 200 might or might not be accepted depending on their filtering responses. But the higher the contrast threshold, the criteria for a pixel becoming a line point becomes more stringent, therefore less line points will be detected.

Note:

If the ‘Dark Line’ is set to 1 (for Picrosirius Red), the low contrast and high contrast will be calculated as and , respectively.

1. **Min/Max Curvature Window**

Twombli quantifies the curvature of ridges/lines in curvature windows bounded by the minimum and maximum curvature windows with a step size of 10.

1. **Minimum Branch Length**

Any line/ridge with a length smaller than this value is ignored.

NOTE: If the minimum branch length value is too low, short branch artefacts are introduced.

1. **Minimum Gap Diameter**

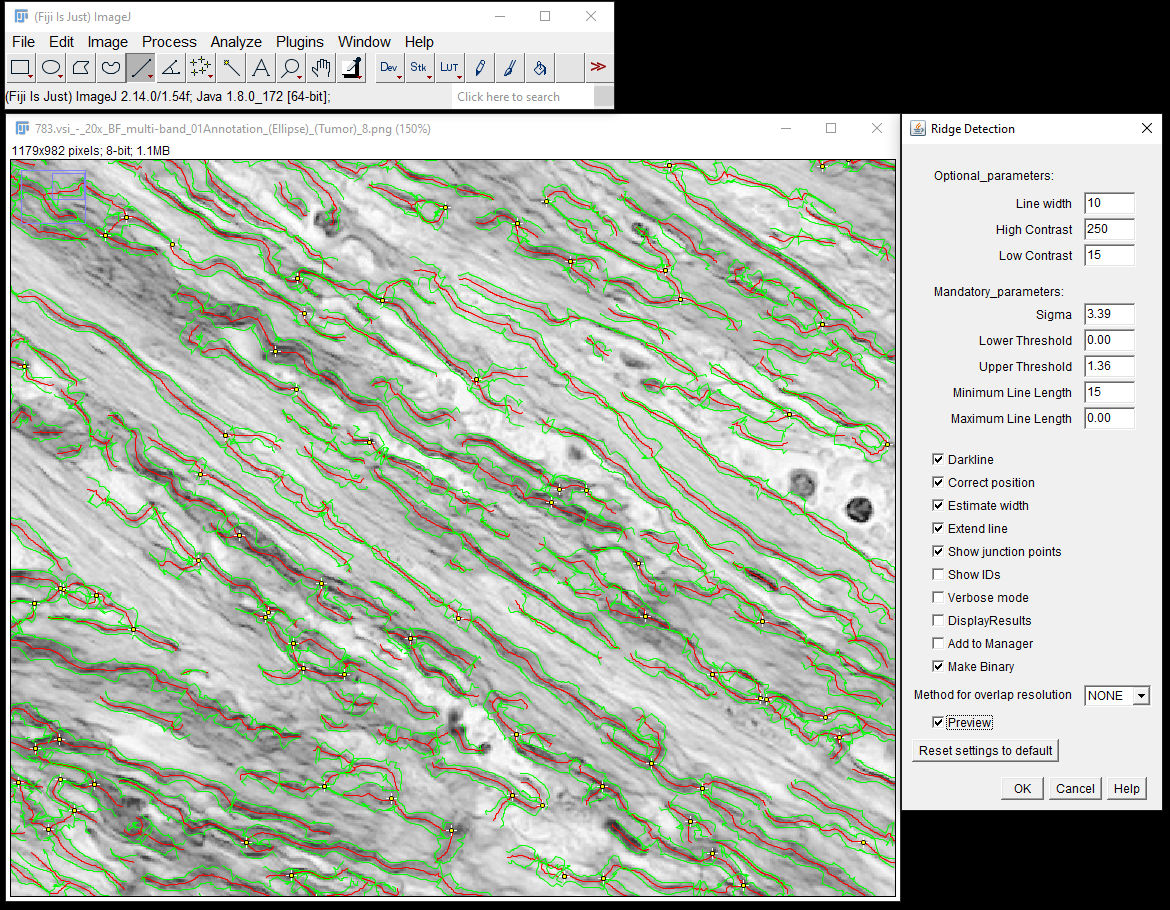
The minimal gap diameter for gap analysis.

If set = 0, gap analysis will not be performed.

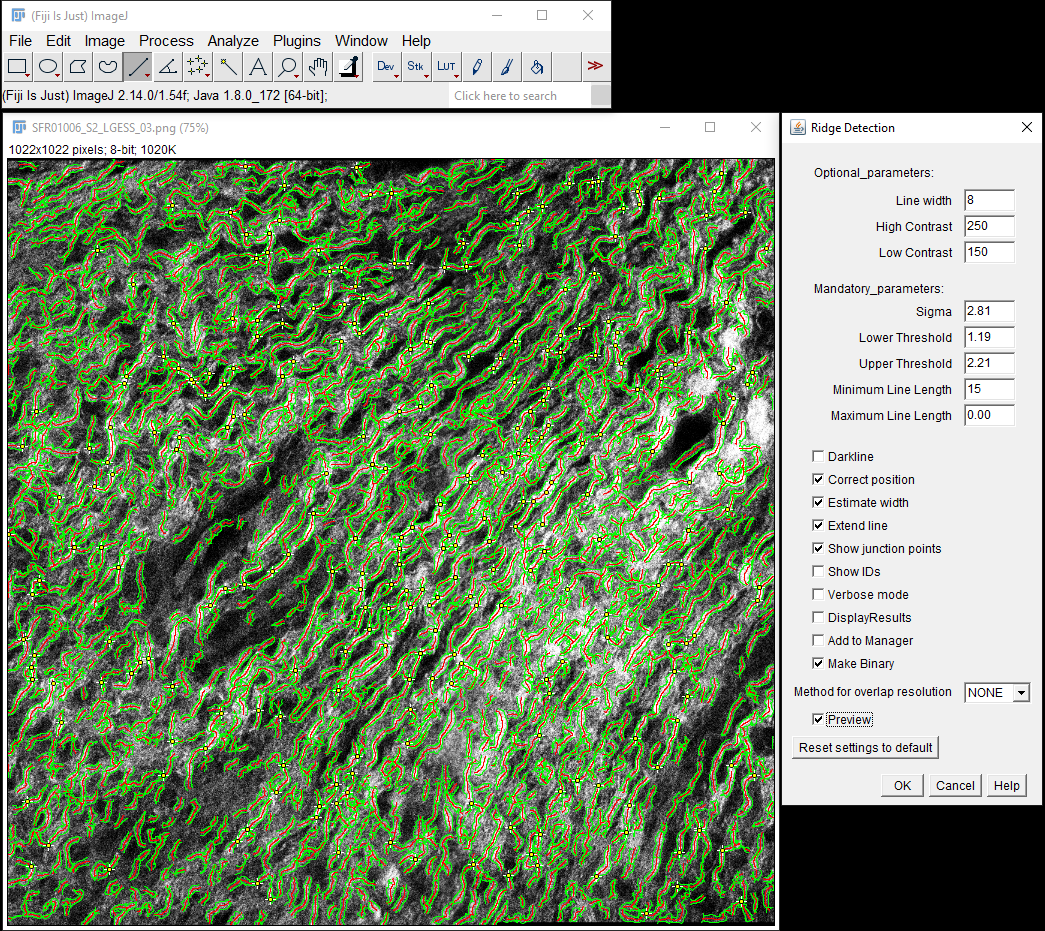
**Estimation of Segmentation and Twombli Parameters**

Before the entire data set is processed, it is recommended to test optimal segmentation and Twombli parameters on a small, representative subset of images.

1. Process images with Segmenter Parameters ‘Mode’ set to ‘segmentation’.
2. Assess results of segmentation and adjust parameters, if necessary, as described above.
3. Open representative images in ImageJ as an 8bit image (grey scale).
4. Ensure that the image resolution is read correctly from the image metadata.
5. Use the line tool to measure the width of a range of collagen fibres in pixels. The min and max width values determine the ‘min line width’ and ‘max line width’ in ‘TwombliParameters.txt’.
6. Use the line tool to measure the minimum and maximum intensity of a range of collagen fibres. This determines the ‘low contrast’ and ‘high contrast’ in ‘TwombliParameters.txt’.
7. Use the line tool to measure the length of a range of short collagen fibre branches in pixels. This determines the ‘minimum branch length’ in ‘TwombliParameters.txt’.

Tip: Use the [Ridge Detection - ImageJ](https://imagej.net/imagej-wiki-static/Ridge_Detection) Plugin to assess min/max line width, low / high contrast and minimum branch length. Images must be converted to 8-bit format first. 

Picrosirius Red



SHG

1. Estimate the minimum gap distance between fibres using the line tool in pixels. This determines the ‘minimum gap diameter’ calculated in the gap analysis.
2. Add determined values to SegmenterParameter.txt and TwombliParameter.txt and save
3. Run entire data set on the desktop Cabana installation or on Google Cloud Cabana.

**Output folder contains following subfolders:**

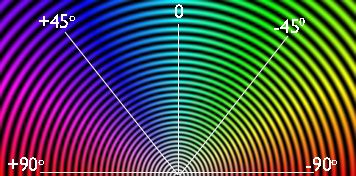
1. **Bins**

Stores the binary masks resulting from ROI extraction. The ROI regions are highlighted in white while backgrounds are highlighted in black.

1. **Colors**

Contains images of the following analysis results:

1. all\_gaps: gap analysis of whole tissue.
2. color\_coherency: orientation coherency/alignment in a local window =2, ~12 x 12 px window, for OrientationJ plugins) of the original image. **Note that the prefix “color” means length visualized in color, not the coherency of color!**
3. color\_length: detected ridges are colour-coded according to fibre length. Note: this only applies to fibres without branches.
4. color\_mask: detected ridges and branches.
5. color\_orientation: fibre orientation in [] within a local window.
6. color\_skeleton: detected fibres with branchpoints (yellow) and endpoints (green).
7. colour\_width: detected ridges with calculated fibre widths.
8. intra\_gaps: gap analysis of intra-collagen fibre gaps.
9. orient\_color\_survey: color coding of orientation in HSB color space, where hue is orientation, saturation is coherency, brightness is the grayscale of the original image.



[Circular color map coding for orientation](https://bigwww.epfl.ch/demo/orientationj/)

1. orient\_vf\_constant: vector field visualization of orientation with constant/equal weights
2. orient\_vf\_wgts\_coherency: vector field visualization of orientation, with vector lengths weighted by coherency.
3. orient\_vf\_wgts\_energy: vector field visualization of orientation, with vector lengths weighted by energy, i.e., gradient vector magnitude.
4. **Eligible**

stores the images to be analyzed after the removal over-sized images. The 'Ignored\_images.txt' file inside this subfolder records the names of the over-sized images that have been ignored.

Note: images that are larger than 2048 x 2048 pixels will automatically be cropped into subregions < 2048 x 2048 px and saved as ROIs.

1. **Exports**

Contains colour-code images of following results:

1. Coherency: coherency/alignment of orientation in a local neighbourhood
2. Colour survey: same as above in 2.i
3. Curve map: curvatures of the detected fibres for selected curvature window sizes
4. Energy: magnitude of gradient vector (
5. Map of tissue gaps (red) – based on full image
6. Map of intra-fibre gaps (green) – based on segmentation ROI
7. Length map: higher intensity values represent longer fibres. The image intensities are stored as unsigned 16-bit integers. Best viewed in ImageJ/FIJI.
8. Mask: detected fibres and fibre branches
9. Orientation: orientation angles in radians ranging from to .
10. Skeleton: shows branchpoints (yellow circles) and endpoints (red circles) of detected fibre spines
11. Width: ridge detection results with the estimated ridge width
12. **HDM**

stores the high-density matrix areas generated by Twombli

1. **Masks**

stores the results of ridge detection, anamorf, and gap analysis in the 'GapAnalysis' folder.

1. **Masks/GapAnalysis**

stores the gap analysis outcomes, comprising of two output images for each image:

One image features red circles, visualizing all gaps, irrespective of whether they occur between collagen fibres or in tissue gaps.

The other image shows green circles depicting gaps within collagen fibre areas (intracollagen gaps).

1. **Ridges**

stores the visualization results of ridge detection as an overlay with the original image.

1. **ROIs**

stores the extracted ROIs.

1. **Twombli\_Results\_Final.csv**

contains all the resultant image statistics.

1. **Version\_params.txt**

Contains selected data of Cabana run, Segmenter and Twombli parameters, Cabana, Bioformats Plugin, IJ ridge detection plugin and Anamorf version numbers.

**Error messages**

1. If you see the message "Structure too complex to find longest path in a reasonable time - skipping," it means that the algorithm has ignored fibers longer than 5000 to avoid excessive running time. If you only see a few of these warning messages for an image, it should not significantly affect the final statistical measurements.
2. Please note that the default allowable image size is 2048\*2048 px, but an image may also be rejected if it contains too much dark background (i.e., the percentage of pixels less than 5 is larger than 99%), as the algorithm may determine that there are insufficient regions of interest to analyse.
3. If a detected ridge structure is too complicated, it normally has many branch- and endpoints, so if the sum of branching and end points is larger than the maximum number (5000) (criteria used by the algorithm to determine if a ridge is too complicated or not), the ridge will be ignored to avoid prolong processing time. So, for long ridges, processed by the Cloud version, their statistics might not be reflected in the final results.
4. If Cabana crashes before the processing has been completed, it can be restarted to continue at the last processed image. Cabana automatically generates a checkpoint file to which it refers when the process is restarted.
5. Images are processed in batches of 5 at a time. While this may add a small additional computational load, it enables the user to resume from the previous batch in case of any errors.

**LIST OF CABANA READ-OUTs (Twombli\_Results\_final.csv)**

Values referring to *WIDTH*, *ROI* and *HDM* are based on the following outputs:

* WIDTH= value based on ridge detection (images named ‘\*\_Width.png’ in “Exports” folder)
* ROI= value based on segmented image (images in “ROI” folder)
* HDM = values based on the high-density matrix (HDM) (Non-black areas/pixels of images in “HDM” folder)

|  |  |  |
| --- | --- | --- |
| **Result** | **Meaning** | **Calculated how?** |
| **Projected Area Ridge Spines (microns^2)** | Projected area of the spines of detected ridges, often one pixel wide. | Ridge spine length in pixels \* Image Res. (microns/pp)^2. |
| **Total Image Area (microns^2)** | Total image area in microns^2 | Total number of pixels \* Image Res. (microns/pp)^2 |
| **% High Density Matrix** | % of image area covered by HDM | High-density matrix areas are obtained using the following three steps:   1. Set pixels outside the range [0, maxDisHDM] to 0, where maxDisHDMis the value of “Maximum Display HDM” parameter specified in TwombliParameters.txt 2. Apply contrast enhancement to the resultant image using the “Contrast Saturation” parameter specified in TwombliParameters.txt 3. High Density Matrix areas are defined by the non-black pixels in the contrast enhanced image.   % HDM is calculated as the portion of pixels in HDM areas. |
| **Avg Thickness (HDM, microns)** | Average fibre thickness calculated based on HDM | Area of High Density Matrix /  total fibre length |
| **Fibre Area (WIDTH, microns^2)** | Total area of fibres calculated based on ridge detection | Width area in pixels \* Image Res. (microns/pp)^2. |
| **Fibre Coverage (WIDTH/Total)** | Normalised fibre area based on ridge detection (normalised to total area) | Fibre Area (WIDTH) / Total area |
| **Avg Thickness (WIDTH, microns)** | Average fibre thickness based on ridge detection | Number of (Width) pixels / total length of fibres |
| **Lacunarity** | Measure of number and size of gaps in matrix.  Larger values indicate larger gaps. Result based on ridge detection. | Calculated as , where *s* and are the standard deviation and mean of the entire binary ridge image, respectively. |
| **Normalised Lacunarity** | Normalised to total image area, resulting a value in [0, 1] | (Lacunarity - 1) / ((1/area ratio) - 1), where area ratio is the ratio of the number of ridge pixels to the total number of pixels. Refer to <https://sci-hub.mksa.top/10.1016/j.jsg.2010.08.010> for more detail. |
| **Total Length (microns)** | Total length of fibres  Based on ridge detection | Total fiber length in microns. Please note it is highly related to the Projected Ridge Area. Their key difference lies in that, total fibre length is derived from pruned ridges (where some artifactual points or tiny branches are removed), whereas the Projected Ridge Area is based on the unpruned ridges, i.e., the direct output of ridge detection. |
| **Avg Length (microns)** | Average length of fibres | Total fibre length / number of endpoints and branchpoints |
| **Endpoints** | Number of endpoints |  |
| **Normalised Endpoints** | Number of endpoints normalised to total fibre length | Number of endpoints / total ridge length in microns |
| **HGU (microns)** | Hyphal growth unit (number of endpoints / 1 µm of fibre length) | Total fibre length / number of endpoints |
| **Branchpoints** | Number of branchpoints |  |
| **Normalised Branchpoints** | Number of branchpoints normalised to total fibre length | Number of branchpoints / total ridge length in microns |
| **Box-Counting Fractal Dimension** | Values between 1 – 2.  Result based on ridge detection. | Fractal dimension is the indicator of the complexity of structure. It is an indicator for the capability of filling space. A more complex structure has a higher tendency to fill up the whole space, thus a higher fractal dimension.  The space filling tendency is measured as the slope of a linear fit to points represented as where is the box size, and represents the count of boxes containing the structure (i.e. ridges). Typically, spans from 2 to one-quarter of the image's largest dimension. Refer to <https://imagej.nih.gov/ij/plugins/fraclac/FLHelp/BoxCounting.htm> for more information. |
| **Curvature** | Mean angle change across the user defined window (ie 20px, 30px etc)  Curvature\_20 = curvature in 20px sliding window |  |
| **Mean Fibre intensity (HDM)** | Mean intensity of either red (PicRed) or grey (SHG) pixels based on HDM |  |
| **Mean Fibre intensity (ROI)** | Mean intensity of either red (PicRed) or grey (SHG) pixels based on ROI |  |
| **Mean Fibre intensity (WIDTH)** | Mean intensity of either red (PicRed) or grey (SHG) pixels based on ridge detection |  |
| **Alignment** | Measure of coherency of fibre orientation in the entire image. | Orientation coherency is calculated as where and are respectively the largest eigenvalue (major axis) and smallest eigenvalue (minor axis) of the global gradient structure tensor. Refer to the [source code](https://github.com/Biomedical-Imaging-Group/OrientationJ/blob/master/src/main/java/OrientationJ_Dominant_Direction.java#L102) for more info. |
| **Mean (Total gap area in microns^2)** | Mean area covered by Max Inscribed Circles in total image | Refer to [here](https://imagej.net/plugins/max-inscribed-circles) for more info about Max Inscribed Circles. |
| **Normalised Mean (Total gap area)** | Max Inscribed Circle area normalised to total image area |  |
| **Std (Total gap area in microns^2)** | StDev of Max Inscribed Circle area in total image (“variability measurement of the Max Inscribed Circles”) |  |
| **Normalised Std (Total gap area)** | Normalised StDev of Max Inscribed Circle area to total gap area |  |
| **Mean (Total gap radius in microns)** | Mean Max Inscribed Circle radius in total image |  |
| **Normalised Mean (Total gap radius)** | Mean Max Inscribed Circle radius normalised to the square root of the total image area |  |
| **Std (Total gap radius in microns)** | StDev of Max Inscribed Circle radius in total image (“variability measurement of the Max Inscribed Circle radii”) |  |
| **Normalised Std (Total gap radius)** | Normalised StDev of gap radius to the square root of the total image area |  |
| **Gaps number (Total)** | Total number of Max Inscribed Circles |  |
| **Gap density (Total, number/microns^2)** | Number of gaps per |  |
| **Mean (ROI gap area in microns^2)** | Mean Max Inscribed Circle area covered by gaps in ROI |  |
| **Normalised Mean (ROI gap area)** | Max Inscribed Circle area normalised to fibre area in ROI |  |
| **Std (ROI gap area in microns^2)** | StDev of Max Inscribed Circle area in ROI (“variability measurement of the Max Inscribed Circle areas”) |  |
| **Normalised Std (ROI gap area)** | Normalised StDev of Max Inscribed Circle area to fibre area in ROI |  |
| **Mean (ROI gap radius in microns)** | Mean Max Inscribed Circle radius in ROI |  |
| **Normalised Mean (ROI gap radius)** | Mean Max Inscribed Circle radius normalised to the square root of the ROI area |  |
| **Std (ROI gap radius in microns)** | StDev of Max Inscribed Circle radii in ROI (“variability measurement of the Max Inscribed Circle radii”) |  |
| **Normalised Std (ROI gap radius)** | Normalised StDev of Max Inscribed Circle radii to the square root of the ROI area |  |
| **Gaps number (ROI)** | Total number of Max Inscribed Circles in ROI |  |
| **Gap density (ROI, number/microns^2)** | Number of Max Inscribed Circles per in ROI |  |
| **Image Res. (microns/pp)** | Image resolution | Read out from image metadata.  NOTE:  Although it is feasible to analyze images of various pixel resolutions, it is important to note that ridge detection outcomes can exhibit significant variations when applied to images with different pixel resolutions. To enhance the comparability of analysis results, it is strongly recommended to work with images that have consistent pixel resolutions. |