MARK FRICKER

AnalyzER

PLANT SCIENCES, OXFORD
FIRST EDITION

PUBLISHED BY PLANT SCIENCES, OXFORD
FIRST EDITION

First printing, July 2018

Current version, December 2018

Contents

Acknowledgements

This work was supported by a Visiting Fellowship at The Institute of Advanced Studies in Durham, The Leverhulme Trust (RPG-2015-437) and The Human Frontier Science Program (RGP0053/2012).

The ER network analysis package is based on the phase congruency analysis originally developed for fungal and slime mold networks with Boguslaw Obara, University of Durham, available at:
http://community.dur.ac.uk/boguslaw.obara/research/software/

This package also uses the following matlab implementations:

Bioformats package for file import:
http://www.openmicroscopy.org/site/products/bio-formats

The phase congruency and skeleton extraction by Peter Kovesi, University of Western Australia:
http://www.peterkovesi.com/matlabfns/

The anisotropic second-order Gaussian kernels by Carlos Lopez-Molina, Universidad Pública de Navarra, Pamplona:
http://www.kermit.ugent.be/software.php?navigatieId=0&categorieId=17

The Frangi 'Vesselness' filter by Marc Schrijver and D. Kroon, University of Twente:
https://uk.mathworks.com/matlabcentral/fileexchange/24409-hessian-based-frangi-vesselness-filter

The export fig package by Oliver Woodford and Yair Altman:
https://uk.mathworks.com/matlabcentral/fileexchange/23629-export-fig

The Matlab implementation of the Boost Graph Library by David Gleich:
https://uk.mathworks.com/matlabcentral/fileexchange/10922-matlabbgl

The Bresenham line drawing algorithm by Aaron Wetzler, Haifa Technion, Israel:
https://uk.mathworks.com/matlabcentral/fileexchange/28190-bresenham-optimized-for-matlab

The CircStat toolbox by Philipp Berens and Marc J. Velasco, tuebingen:
P. Berens, CircStat: A Matlab Toolbox for Circular Statistics, Journal of Statistical Software, Volume 31, Issue 10, 2009. http://www.jstatsoft.org/v31/i10
https://uk.mathworks.com/matlabcentral/fileexchange/10676-circular-statistics-toolbox--directional-s

1
Introduction

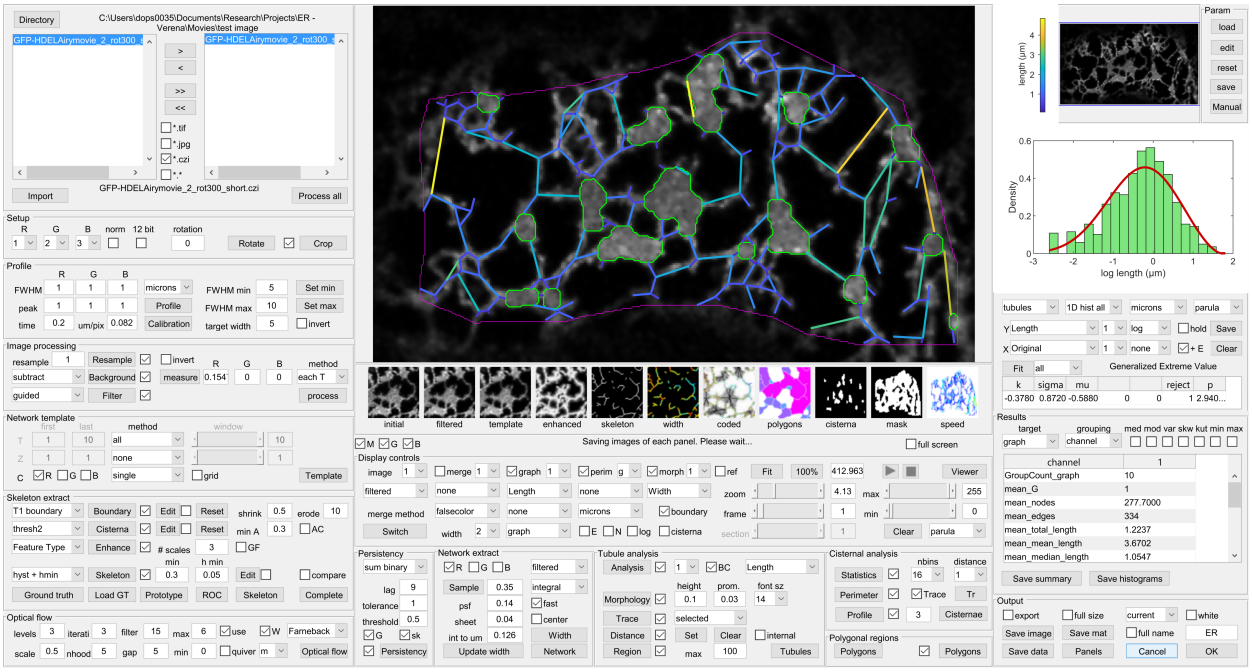


Figure 1.1: The GUI interface for the AnalyzER network analysis program

1.1 Overview

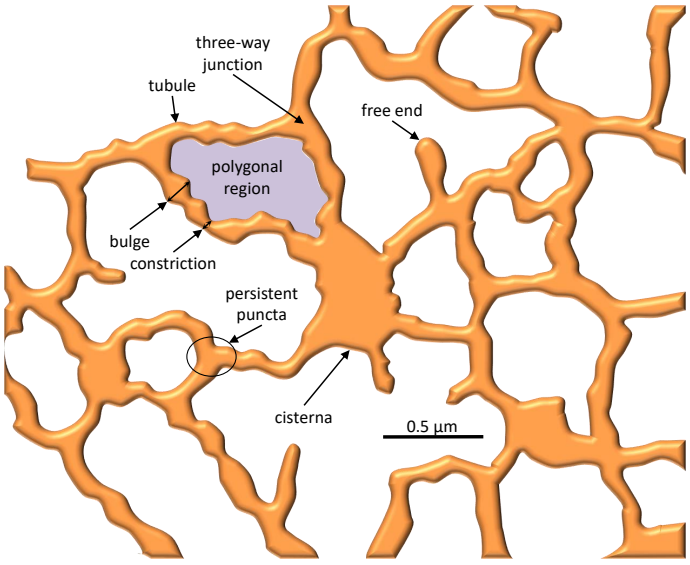
The endoplasmic reticulum (ER) forms a complex and dynamic network of tubules and sheet-like cisternae that ramify throughout the cytoplasm (Fig. 1.2), see Westrate *et. al.* (2015)[1]. The aim of the ER network analysis program is to quantify:

- The length, width, morphology and protein localisation along ER tubules
- The size, shape, and protein distribution within and around the perimeter of the ER cisternae
- The topological organisation of the tubular and cisternal network determined using graph-theoretic metrics

[1] L. M. Westrate, J. E. Lee, W. A. Prinz, and G. K. Voeltz. Form follows function: The importance of endoplasmic reticulum shape. *Annual Review of Biochemistry*, 84:791–811., 2015

- The local speed and direction of movement of tubules and cisternae
- The presence and distribution of immobile nodes, tubules and cisternae using persistency mapping
- The size and shape of the polygonal regions enclosed by the network

Figure 1.2: Schematic representation of typical ER morphology



The programs were designed to quantify ER organisation in confocal optical sections of plant epidermal cells, where the ER is confined to a very thin layer of cytoplasm appressed to the periclinal cell wall as a planar, 2-D network, but can also be applied to peripheral ER in spreading animal cells grown in culture, where the ER is also constrained to a 2-D network. The input image typically comprises a single plane (x,y) confocal fluorescence image of ER-targeted fluorescent protein; a single plane multi-channel (x,y,λ) image or a multi-channel time-series (x,y,λ,t). The network

analysis progresses through a number of parallel threads that are designed to extract different information from the underlying image to characterise the tubular network, cisternae and enclosed polygonal regions. A flow diagram of the overall sequence for morphological measurements for a single (x,y) plane is shown in Fig. 1.3.

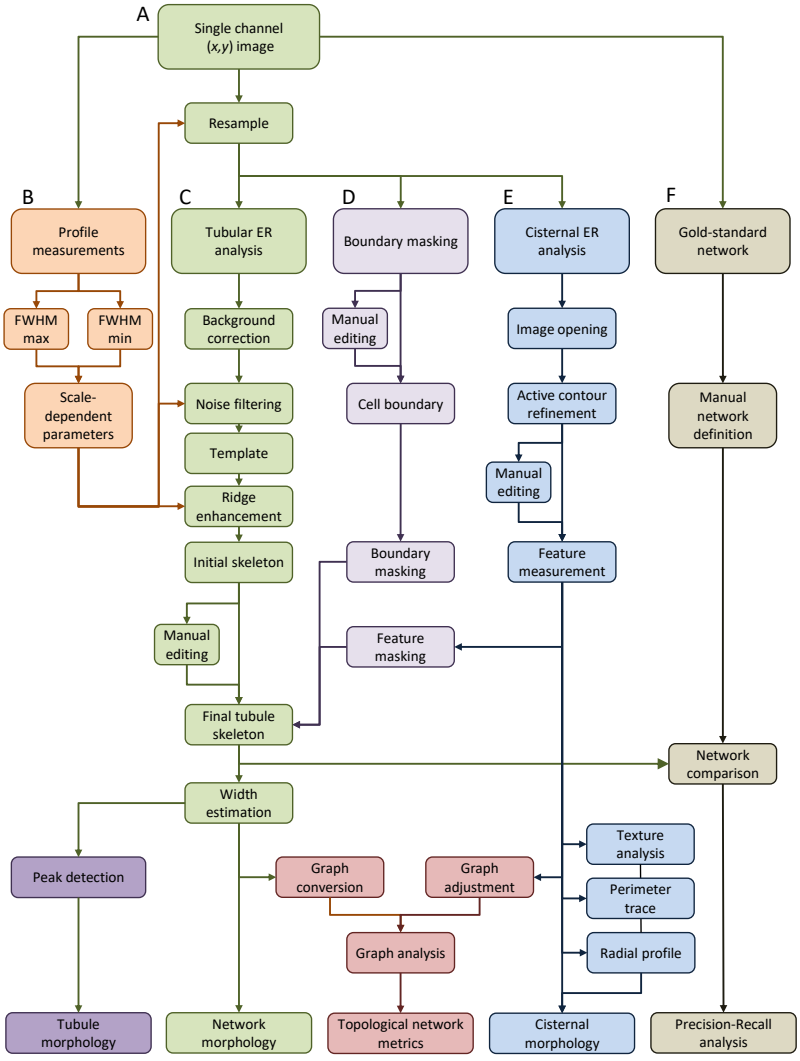


Figure 1.3: A flow diagram showing the main steps in the ER analysis. (A) The starting point is typically a single channel confocal fluorescence of the ER labelled with a fluorescent protein. (B) The minimum and maximum tubule diameters are estimated manually from line transects to resize the image and standardise all the subsequent processing parameters; (C) ER tubules are segmented following filtering and enhancement steps to give a single pixel wide skeleton. This provides basic morphological information on the length and width of the tubules, and can be interrogated further to examine individual tubule morphology. (D) The analysis can be constrained to a particular cell or sub-cellular region by masking the image; (E) ER cisternae are detected independently using image opening followed by active contour refinement; (F) the performance of the automated segmentation approaches can be compared to a ground-truth pixel skeleton defined manually

1.2 *Quantitation of network dynamics*

The plant ER is highly dynamic, with much of the network being continuously re-modelled with regions of tubule growth, shrinkage, or lateral sliding, expansion, contraction and movement of cisternae, and highly dynamic streams running along actin cables[2]. To capture some of these dynamics from time-series images, network extraction and analysis can be repeated for each frame in a time-series to extract time-dependent changes. Analysis of a multi-channel time-series follows the same pattern as a single channel

[2] L. R. Griffing, C. Lin, C. Perico, R. R. White, and I. Sparkes. Plant ER geometry and dynamics: biophysical and cytoskeletal control during growth and biotic response. *Protoplasma*, 254: 43–56, 2017

time-series image, but also allows comparison between the different channels, usually as a ratio or covariance measure (Fig. 1.4).

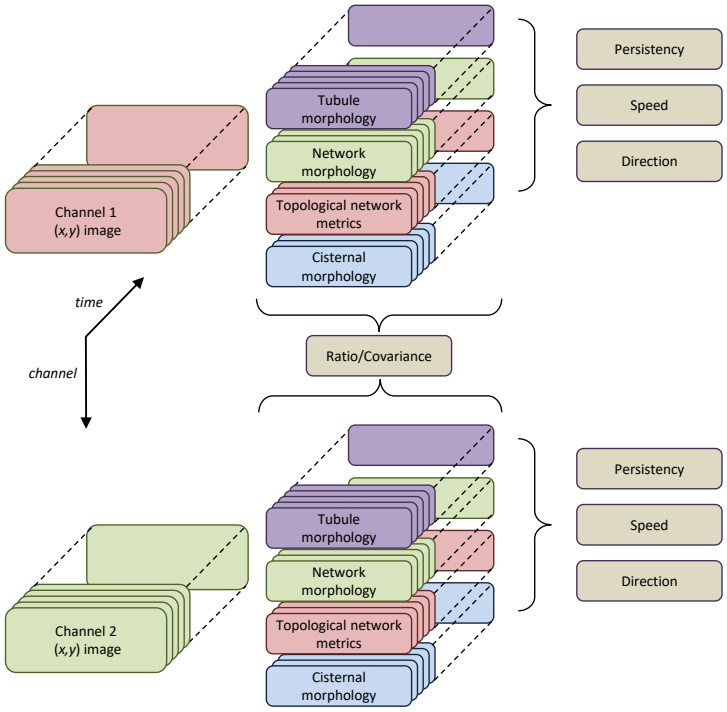


Figure 1.4: Measurement of ER structure and dynamics in multi-channel time-series images to include time-varying metrics, such as persistency, speed and direction

The local velocity of tubules and cisternae can be quantified through the time-series using optical flow techniques[3] (see Chapter 7).

The plant ER network also has a subset of static elements, such as ER-plasma membrane contact sites (EPCS), that remain in a fixed position for a period of time[4]. The spatial persistence of tubules, cisternae or specific puncta can be used to identify fixed scaffolding points by quantifying parts of the network that remain in position over a given time interval using persistency mapping (see Chapter 8).

[3] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92:1–31, 2011

[4] I. Sparkes, J. Runions, C. Hawes, and L. Griffing. Movement and remodeling of the endoplasmic reticulum in nondividing cells of tobacco leaves. *The Plant Cell*, 21:3937–3949, 2009b

1.3 *Segmentation of the ER network*

The simplest method to identify the ER automatically would be an intensity-based segmentation of the fluorescent image to give a binary image, with ones representing the ER structure and zeros for the background. However, the resultant binary image is critically dependent on the value for the threshold used, and it is rare that a single threshold provides adequate segmentation without either losing dimmer structures if it is set too high, or artificially expanding and fusing adjacent regions if it is set too low[5]. Thus the approach adopted here exploits additional intensity-independent information over a range of scales and orientations to enhance the network structure, prior to segmentation as a single-pixel wide

[5] A.-N. Bouchekhima, L. Frigerio, and M Kirkilionis. Geometric quantification of the plant endoplasmic reticulum. *J. Microscopy*, 234:158–172, 2009

skeleton (Fig. 1.5). Full details of the different methods are given in Chapter 2.

The expected width of the ER tubule is ranges from 30 nm in yeast to 50-70 nm in plant and animal cells (Westrate *et. al.*, 2015), which is below the resolution of the confocal microscope, but can just be resolved with super-resolution techniques, such as stimulated emission depletion microscopy (STED[6]). Nevertheless, for most laboratories, access to super-resolution techniques may be limited, necessitating the development of approaches that can be used on a routine basis with existing tools, particularly for dynamics imaging of ER movement in living tissues[7]. In principle, the width of the ER can be estimated from the intensity information, even if this is below the resolution limit of the microscope system, with assumptions about the distribution of the fluorescent lumenal marker and the point-spread-function (psf) of the microscope[8]. In the approach adopted here, the skeleton is used as a template to interrogate the image locally to provide an estimate of the relative amount of fluorescent probe present, and the width is inferred from the integrated intensity signal.

Topological measures of the ER network structure can also be extracted following conversion of the pixel skeleton to a weighted, undirected graph, where nodes represent junction points and edges represent the tubules that connect them (Fig. 1.6). Identification of nodes and edges also allows assignment of ER metrics (length, width, velocity, persistence) to individual tubules. The metric can be displayed as colour-coded edges superimposed on the image, or mapped back into the pixel skeleton.

Unlike morphological measurements, the topology of the network is less sensitive to the resolution of the imaging system as it reflects the connectivity of the ER rather than the physical size of the components (Bouchekhima *et al., 2009*).

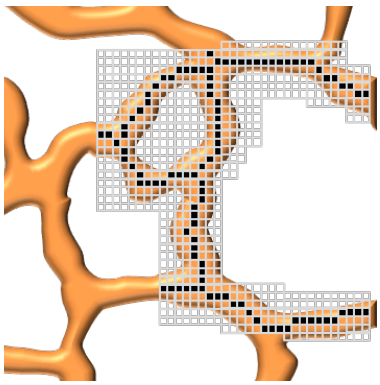


Figure 1.5: Conversion of the ER image to a single-pixel wide skeleton

[6] B. Hein, K. I. Willig, and S. W. Hell. Stimulated emission depletion (sted) nanoscopy of a fluorescent protein-labeled organelle inside a living cell. *PNAS*, 105:14271–14276, 2008

[7] T.J. Lambert and J.C. Waters. Navigating challenges in the application of superresolution microscopy. *J. Cell Biol.*, jcb.201610011, 2016

[8] G. J. Streekstra and J. van Pelt. Analysis of tubular structures in three-dimensional confocal images. *Network: Computation in Neural Systems*, 13: 381–395, 2002

1.4 *Analysis of the ER cisternae*

The ER cisternae are typically segmented and analysed separately to give morphological measures, such as their area and shape parameters (see Chapter 11). In addition, the distribution of different ER-shaping proteins can be tracked along the perimeter of the cisternae, or analysed in radial profiles across the cisternal boundary (Fig. 1.7). When the network is converted to a graph representation, the ER cisternae are represented as a 'super-node' placed at the weighted centroid and connected to all the tubules incident on the boundary.

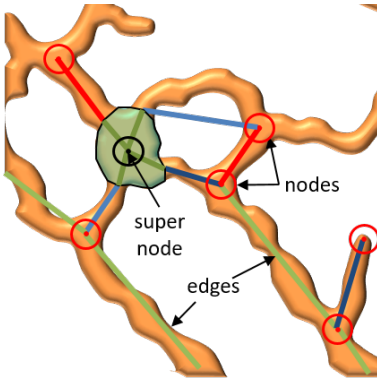


Figure 1.6: Conversion of the ER skeleton to a weighted graph

Recently it has been proposed that many sheet-like regions, that would previously have been identified as ER-cisternae with a continuous lumen, may actually result from local appression of multiple tubes to give a sub-resolution tubular matrix that can only be resolved with super-resolution techniques[9]. If this is correct, then tubular markers should also be associated with the tubular

[9] C.J. Nixon-Abell, J.and Obara, A.V. Weigel, D. Li, W.R. Legant, C.S. Xu, H.A. Pasolli, K. Harvey, H.F. Hess, E. Betzig, C. Blackstone, and J. Lippincott-Schwartz. Increased spatiotemporal resolution reveals highly dynamic dense tubular matrices in the peripheral er. *Science*, 354:aaf3928, 2016

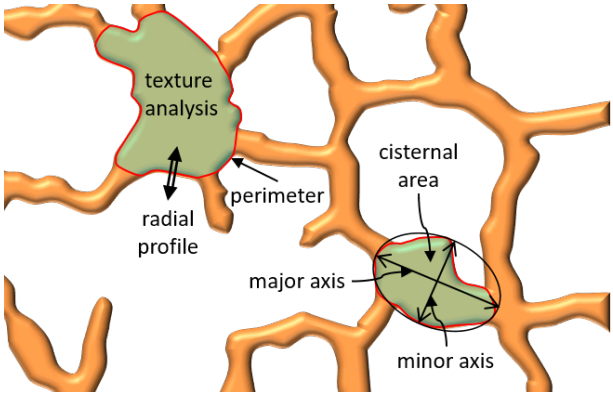


Figure 1.7: Characterisation of ER cisternae through morphological measurements and texture analysis

matrix, providing some texture to the apparent sheet regions. Texture measures can be determined from pairwise comparison of pixel values at varying distances and orientation to give a grey-level co-occurrence matrix (GLCM)[10].

[10] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, pages 610–621, 1973

1.5 *Polygonal regions*

The polygonal regions enclosed by the ER network can also be segmented and their statistical properties measured in much the same way as morphological measurements of the ER cisternae. A complete measurement set would require a full 3-D image and 3-D segmentation methods which is beyond the scope of the current software. Thus at present, only statistics for fully enclosed polygons in the imaging plane are calculated.

2
Approaches to ridge enhancement and segmentation

2.1 *Introduction*

There are a wide range of different approaches that could be used to enhance the ridge-like structures in ER networks. This Chapter provides some of the theoretical background to the approaches implemented in the network analysis package.

2.2 *'Vesselness'*

One of the first methods to identify ridges exploited the local curvature of the intensity landscape as estimated from the Hessian (H_σ), comprising second-order partial derivatives, D_{aa} along direction a, of the intensity image (I), where the value of the standard deviation of the Gaussian kernel (σ) is varied over a range of scales that span the sizes of the underlying features[1], see for example Fig. 2.1A-C:

[1] A. F. Frangi, W. J. Niessen, K.L. Vincken, and M.A. Viergever. *Multiscale vessel enhancement filtering*, pages 130–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998

$$H_\sigma = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} = \begin{bmatrix} \frac{\delta^2 I}{\delta x^2} * G_\sigma & \frac{\delta^2 I}{\delta x \delta y} * G_\sigma \\ \frac{\delta^2 I}{\delta x \delta y} * G_\sigma & \frac{\delta^2 I}{\delta y^2} * G_\sigma \end{bmatrix} \tag{2.1}$$

where

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{2.2}$$

In the resultant scale-space representation, further information on ridge-like features can be extracted from the eigenvalues and eigenvectors of the Hessian, which show characteristic behaviour for a filamentous structure. By ordering the eigenvalues in terms of their absolute magnitude ($|\lambda_1| < |\lambda_2|$, for a 2D image), the smallest eigenvalue ($|\lambda_1|$) denotes the minimum change in intensity, with the corresponding eigenvector oriented along the centreline of the ridge, whilst the largest eigenvalue ($|\lambda_2|$) and eigenvector determine the orientation of the maximum curvature, normal to the ridge centreline. Prominent structures are distinguished from the background by relatively large values of the eigenvalues ($\sqrt{\lambda_1^2 + \lambda_2^2}$). In addition, the ratio $|\lambda_1|/|\lambda_2|$ gives an indication of how blob-like ($|\lambda_1| \approx |\lambda_2|$) or elongated and filament-like ($|\lambda_1| \ll |\lambda_2|$) the structure is at that point. Thus the 'Vesselness' (V_σ)

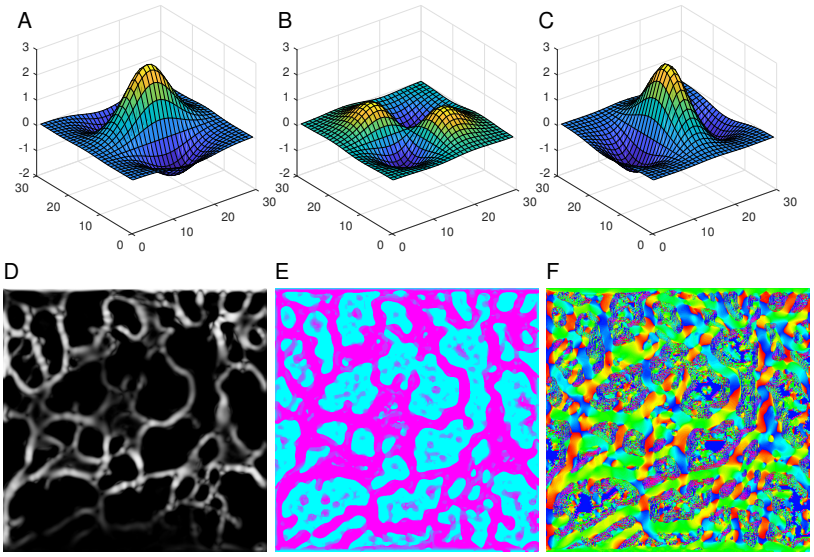


Figure 2.1: Frangi 'Vesselness' filter. (A-C) show surface plots of the D_{xx}, D_{xy} and D_{xy} filters at scale 7 that are used to calculate the second-order derivative of the image. The filters are shown inverted to highlight the shape of the ridge detector; (D) The 'Vesselness' output, calculated over 11 scales; (E) The scale at which the maximum response occurred; (F) the orientation of the maximum response.

measure (Frangi *et al.* 1998), defined by Equation (2.3), is large at those pixels that are part of a linear structure of scale (σ).

$$V_\sigma = e^{-\frac{\lambda_1^2}{2\beta^2\lambda_2^2}}\left(1 - e^{\frac{\lambda_1^2+\lambda_2^2}{2c^2}}\right) \tag{2.3}$$

Note that the relative contributions of the geometric ratio and the intensity components at a given scale (σ) are controlled by the coefficients β and c, respectively. Typically, β is set to 0.5 and c is set to half the maximum Hessian norm. Multi-scale 'Vesselness', for a given set of scales spanning the expected width of the vessels, can be computed as the maximum of the 'Vesselness' values calculated at each scale (Fig. 2.1D,E), and the eigenvectors at that scale used to define local orientation (Frangi *et al.* 1998).

Note: whilst contrast for the tubular regions is improved, only the edges of the cisternal regions are retained in the enhanced image.

2.3 *'Neuriteness'*

An alternative weighting of the eigenvalues of the Hessian matrix was proposed by Meijering *et al.* (2004)[2]:

$$H' = \begin{bmatrix} D_{xx} + \alpha D_{yy} & (1-\alpha)D_{xy} \\ (1-\alpha)D_{xy} & D_{yy} + \alpha D_{xx} \end{bmatrix} \tag{2.4}$$

Where α is set to be $-1/3$ such that the filter used in the calculation of the Hessian matrix is maximally flat in its longitudinal direction, effectively generating an anisotropic second order Gaussian filter (Fig. 2.2A). Conveniently, these kernels can be implemented as steerable filters constructed from a set of basis kernels[3]. The 'Neuriteness' measure at scale σ, (N_σ) is determined from the modified

[2] E. Meijering, M. Jacob, J. Sarria, P. Steiner, H. Hirling, and M Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry*, 58: 167 – 176, 2004

[3] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9:891–906, 1991

eigenvalues as:

$$N_\sigma = \begin{cases} \frac{\lambda_\sigma}{\lambda_{\sigma,min}} & \text{if } \lambda_\sigma < 0 \\ 0 & \text{if } \lambda_\sigma \geq 0 \end{cases} \tag{2.5}$$

where λ_σ is the larger in absolute magnitude of the two modified eigenvalues, and $\lambda_{\sigma,min}$ is the smallest value of λ over all pixels such that:

$$\lambda_{\sigma,1}{}' = \lambda_{\sigma,1} + \alpha\lambda_{\sigma,2} \tag{2.6}$$

$$\lambda_{\sigma,2}{}' = \lambda_{\sigma,2} + \alpha\lambda_{\sigma,1} \tag{2.7}$$

$$\lambda_\sigma = \max\left(|\lambda_{\sigma,1}{}'|, |\lambda_{\sigma,2}{}'|\right) \tag{2.8}$$

$$\lambda_{\sigma,min} = \min_{\mathbf{p}\in I}(\lambda_\sigma) \tag{2.9}$$

$\lambda_{\sigma,1}, \lambda_{\sigma,2}$ are the eigenvalues of the Hessian matrix $H_\sigma(\mathbf{p})$, at pixel $\mathbf{p}$, for a given scale parameter σ. The maximum response across all scales gives the 'Neuriteness' image (Fig. 2.2C)

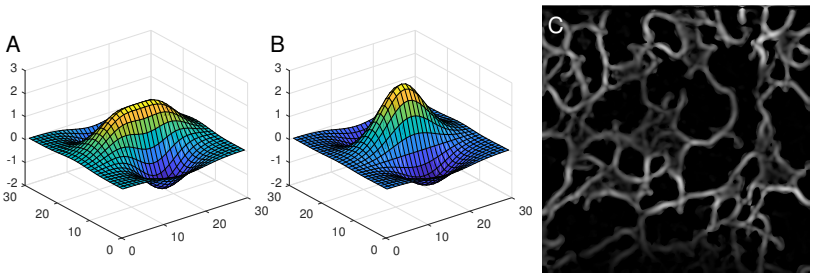


Figure 2.2: Meijering 'Neuriteness' filter. (A) A surface plot for the 'flattened' D_{xx} filter, shown inverted to highlight the shape of the ridge detector; (B) A normal D_{xx} filter for comparison; (C) The 'Neuriteness' output, calculated over 7 scales.

2.4 *Second-order anisotropic Gaussian kernels (SOAGK)*

The use of second-order derivatives of anisotropic Gaussian kernels (SOAGKs) was developed further by Shui *et al.* (2012)[4] and Lopez-Molina *et al.* (2015)[5] to improve detection of ridge like elements. The SOAGK are applied at a range of orientations to give anisotropic directional derivative (ANDD) filters at each scale (Fig. 2.3, A-C). On their own ANDD filters also generate extensions at the end of edge segments, termed edge-stretch, which has the benefit of improving local connectivity by filling in small gaps, particularly at junctions that occur in the 'Vesselness' filter for example, but with the disadvantage of adding spurious features at the end of edge segments. The latter errors can be minimised by using a fused detector that combines the ANDD filter with a small isotropic Gaussian as a geometric mean (Shui and Zhang, 2012). The enhanced edge image is taken as the maximum response at any scale and orientation (Fig. 2.3, E). These filters give strong responses when aligned to the dominant ridge at each scale, and provide estimates of the ridge intensity and ridge orientation without calculation of the eigenvalues and eigenvectors. In addition, the response at

[4] P.-L. Shui and W.-C. Zhang. Noise-robust edge detector combining isotropic and anisotropic gaussian kernels. *Pattern Recognition*, 45:806 – 820, 2012

[5] C. Lopez-Molina, G. V. D. de Ulzurrun, J. M. Baetens, J. Van den Bulcke, and B. De Baets. Unsupervised ridge detection using second order anisotropic gaussian kernels. *Signal Processing*, 116:55–67, 2015

junctions is not attenuated to the same degree as the 'Vesselness' response because of the edge-stretch phenomena. Conversion to a single-pixel wide skeleton then uses local non-maximal suppression to identify key pixels on the ridge centerline, followed by hysteresis thresholding to identify pixels that form the connected skeleton.

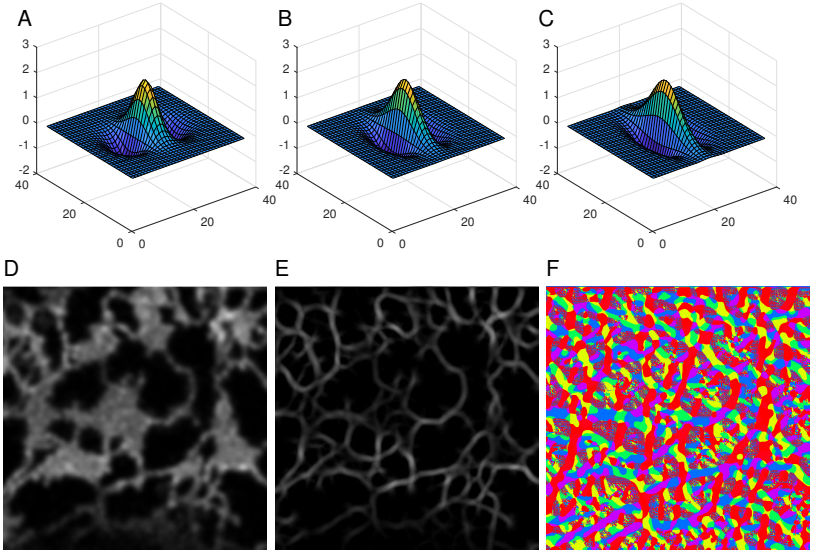


Figure 2.3: Second-order anisotropic Gaussian filter (SOAGK) after Lopez-Molina et al (2015). (A-C) Surface plots for the D_{xx} filter with 1,1.3 and 1.5 levels of anisotropy, shown inverted to highlight the shape of the ridge detector; (D) Original Physarum image (E) Output from the 'SOAGK' filter, calculated over 7 scales; (F) direction of the dominant ridge.

2.5 *Intensity-independent enhancement using phase-congruency*

While ridge enhancement can be built on purely intensity-based filters, such as the Hessian or SOAGKs, these have the downside of being sensitive to changes in image contrast, which often leads to loss of a few pixels from the skeleton during the subsequent thresholding step, effectively disconnecting these edges. This can be ameliorated to some extent by inclusion of a local contrast equalisation step prior to enhancement (Shui and Zhang, 2012), or by the use of adaptive or hysteresis thresholding during segmentation (Lopez-Molina *et al.* 2015). Nevertheless, in these approaches it is critical to establish a reliable, context-dependent threshold selection to achieve segmentation of a fully connected network.

Human observers face a similar challenge when trying to discriminate edges or ridges is a complex visual field. Morrone and Owens (1987)[6] proposed that human edge perception depends on the degree of phase congruency, which is independent of the image brightness. Phase congruency can be estimated from the local energy at each location, determined by convolution of the image with Gabor filters at varying scale and orientation[7]. The phase congruency approach has been developed further as a generic means to extract a variety of image features by Kovesi[8]. Kovesi also introduced a range of improvements to the original measure to improve its overall utility, including log Gabor filters to increase the filter bandwidth, procedures for automated noise rejection, weighting

[6] M.C. Morrone and R.A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303 – 313, 1987

[7] S. Venkatesh and R. Owens. On the classification of image features. *Pattern Recognition Letters*, 11:339–349, 1990

[8] P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1:1–26, 1999

to select against phase congruency of only a few frequencies, and improved spatial precision by including both the cosine and sine of the phase in the estimate. These provide good ridge enhancement, irrespective of image intensity, but also increase the number of parameters that can be tuned to achieve the best enhancement in any particular context.

Following Kovesi, the local energy for a one-dimensional profile, $I(x)$ is given by:

$$E(x) = \sqrt{F^2(x) + H^2(x)} \tag{2.10}$$

where $F(x)$ is the signal with its DC component removed, and $H(x)$ is the Hilbert transform of $F(x)$, obtained by convolving the signal with a quadrature pair of log Gabor filters at scale n, and summing the even filter convolutions $(e_n(x))$ to give $F(x)$, and the odd filter convolutions $(o_n(x))$ to give $H(x)$. The phase congruency $PC(x)$, is normalised by the sum of the Fourier amplitudes $\sum_n A_n(x) \simeq \sum_n \sqrt{e_n(x)^2 + o_n(x)^2}$, with the addition of a small constant ε to improve stability at low Fourier amplitudes:

$$PC(x) = \frac{E(x)}{\sum_n A_n + \varepsilon} \tag{2.11}$$

The log Gabor filter bank is controlled by the minimum wavelength scale, the number of scales, the frequency bandwidth, and the number of filter orientations. Each of these requires some optimisation to achieve good enhancement for specific types of biological networks (Fig. 2.4B).

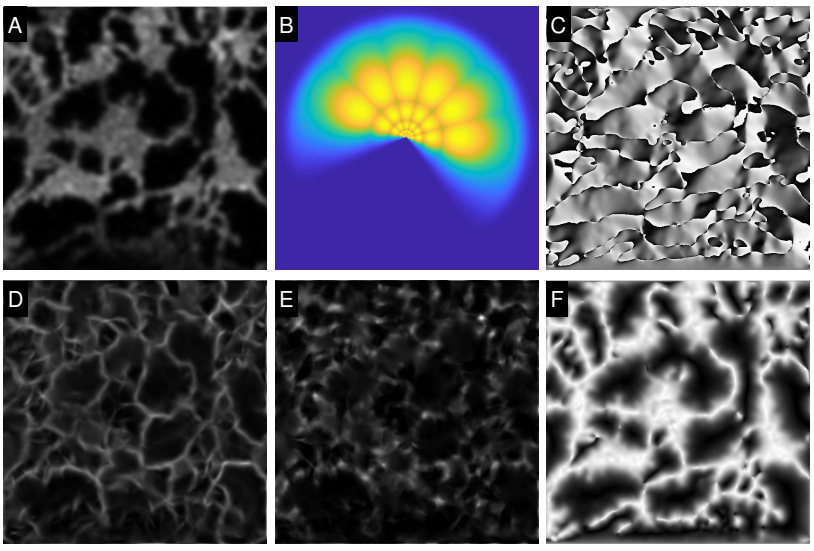


Figure 2.4: Phase Congruency filter. (A) Original image; (B) Set of log Gabor filters in the Fourier domain for 6 scales and 6 orientations; (C) Orientation image from the maximum response; (D) Maximum moment from the phase-congruency filter as a measure of edge strength of the tubules; (E) Minimum moment from the phase-congruency response, which highlights junctions; (F) Local weighted mean phase angle ('Feature Type'), which provides a robust output for segmentation.

The next tuneable parameter controls the amount of noise rejection. To estimate the amount of noise adaptively from the image, Kovesi used a measure of the mean (μ_R) and variance (σ_R^2) of the Rayleigh distribution (R) describing the noise distribution at the smallest scale, with the assumption that ridges are relatively

sparsely distributed in the image, so the mean at this scale will be dominated by background noise. Thus the noise threshold (T), is given by the mean noise response plus some number, k, of deviation units:

$$T = \mu_R + k\sigma_R \tag{2.12}$$

The local energy term $E(x)$ is therefore modified by subtracting the estimated noise (and setting any values below zero to zero).

The second set of tuneable parameters relate to the minimum spread of frequencies required to constitute a useful estimate of phase congruency. In the case of the ER network we are concerned with the detection of ridges, rather than lines or step functions. The expected power spectrum of a ridge falls off at $1/\omega^4$, where ω is the centre frequency of the filter, which gives an expected distribution of frequency responses strongly skewed towards low-frequency end. The significance of $PC(x)$ can be down-weighted if the spread of frequencies is too narrow, however, in the case of ridge detection, this criterion should not be too harsh. Kovesi provides an estimate of the frequency spread by considering the normalised ratio of the sum of the Fourier amplitudes divided by the maximum response:

$$s(x) = \frac{1}{N}\left(\frac{\sum_n A_n(x)}{A_{max}(x) + \varepsilon}\right) \tag{2.13}$$

Where N is the total number of scales, $A_{max}(x)$ is the maximum filter response at x, and ε prevents division by zero. To penalise regions with few frequency components, the weighting function is constructed as a sigmoidal function:

$$W(x) = \frac{1}{1 + e^{\gamma(c - s(x))}} \tag{2.14}$$

where c is the cut-off value below which phase congruency values are penalised, and γ is a gain factor that controls the sharpness of the cut-off

$$PC(x) = \frac{W(x)\lfloor E(x) - T \rfloor}{\sum_n A_n(x) + \varepsilon} \tag{2.15}$$

Where $\lfloor\ \rfloor$ denotes that $E(x) - T$ is equal to itself for positive values and zero otherwise.

The final amendment that Kovesi proposes is to include the information from both the cosine of the phase deviation, which should be large if phase congruency is high, and the absolute value of the sine of the phase deviation, which should be small (2.16):

$$\Delta\Phi_n(x) = \cos(\phi_n(x) - \bar{\phi}(x) - |\sin(\phi_n(x) - \bar{\phi}(x)| \tag{2.16}$$

This gives the complete estimate of phase congruency as:

$$PC(x) = \frac{\sum_n W(x)\lfloor A_n(x)\Delta\Phi_n(x) - T \rfloor}{\sum_n A_n(x) + \varepsilon} \tag{2.17}$$

The two dimensional extension of the phase congruency gives:

$$PC(x) = \frac{\sum_o \sum_n W_o(x) \lfloor A_{no}(x) \Delta\Phi_{no}(x) - T_o \rfloor}{\sum_o \sum_n A_{no}(x) + \varepsilon} \qquad (2.18)$$

Where o is the index over orientations.

3
Installation

3.1 Overview

All the programs were written in MATLAB (The Mathworks, Nantick) and are packaged in a single compiled executable file for distribution as a standalone package, or as an app that runs within the MATLAB environment. The program, manual and tutorial can be downloaded from the Oxford Research Archive:

`https://ora.ox.ac.uk/`

3.2 Installation of the stand-alone program

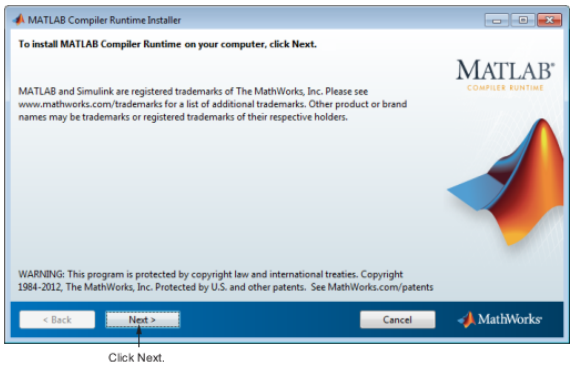
The software has been tested on Windows 10, and requires a minimum screen resolution of 1600 x 900. In addition, an appropriate version of the MATLAB Compiler Runtime (MCR) is required to install the set of shared libraries that enables execution of the compiled MATLAB application. The MCR should automatically download from the MathWorks website when the program is installed for the first time. Alternatively MCR can be downloaded from the MathWorks Website:

`http://www.mathworks.com/products/compiler/mcr`.

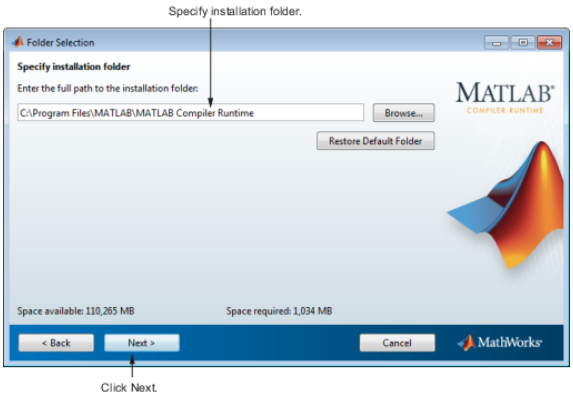
To install the MCR and standalone package, double-click the compiled MATLAB self-extracting archive file. This extracts the MATLAB Runtime Installer from the archive, along with all the files that make up the deployed MATLAB environment. Once all the files have been extracted, the MATLAB Runtime Installer starts automatically. When the MATLAB Runtime Installer starts, it displays the following dialog box. Read the information and then click **Next** to proceed with the installation.

Specify the folder in which you want to install the MATLAB runtime in the Folder Selection dialog box and click **Next**. It is recommended to keep the default settings as this ensures the path to other program files is set automatically.

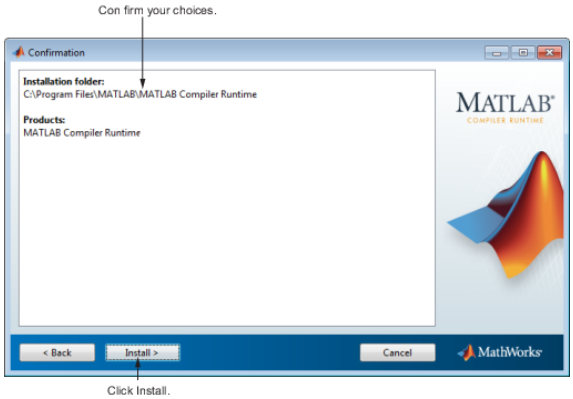
Note: On Windows systems, you can have multiple versions of the MATLAB runtime on your computer, but only one installation for any particular version. If you already have an existing instal-



lation, the MATLAB runtime Installer does not display the Folder Selection dialog box because you can only overwrite the existing installation in the same folder.



Confirm your choices and click **Install**. The MATLAB Runtime Installer starts copying files into the installation folder



Click **Finish** to exit the installer.

MATLAB Runtime Installer Readme File: A readme.txt file is included with the MATLAB Runtime Installer. This file, visible when the MATLAB Runtime Installer is expanded, provides more detailed information about the installer and the switches that can be used with it.

3.3 *Installation of the MATLAB app*

The MATLAB® app installer file, .mlappinstall, contains everything necessary to install and run an app within the MATLAB® environment, including the source code, supporting data, information (such as product dependencies), and the app icon. Double-clicking on the .mlappinstall file should launch MATLAB if it is not already running, and install the app in the app toolbar. the program can be run, by clicking on the icon in the toolbar.

The program uses the following toolboxes:

- 'Signal Processing Toolbox'
- 'Image Processing Toolbox'
- 'Statistics and Machine Learning Toolbox'
- 'Curve Fitting Toolbox'
- 'Bioinformatics Toolbox'
- 'Computer Vision System Toolbox'

Note: Some functions may not be available in the app as they require pre-compiled components such as MEX files, that are not packaged in the app.

3.4 *Installation of additional program files needed*

A number of additional files needed to run the full suite of programs may also be installed at the same time as the main program. The *Bio-Formats* package (Linkert et al. 2010[1]) has been designed to read in images from different microscope manufacturers and store them in a standardised format and should be included in the

[1] M. Linkert, C. T. Rueden, C. Allan, J. M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. Macdonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. W. Eliceiri, and J. R. Swedlow. Metadata matters: access to image data in the real world. *J Cell Biol*, 189:777–82, 2010

program directory during installation. Full details are available on the open microscopy website:

`http://www.openmicroscopy.org/`

If installed separately, the bioformats_packages.jar program needs to be available on the search path or installation directory of the matlab programs. bioformats_package.jar is available from:

`http://downloads.openmicroscopy.org/bio-formats`

The latest version of Java needs to be installed, and is available from:

`http://www.java.com/en/`

Output of images at full resolution uses *export_fig.m* originally written by Oliver Woodford (2008-2014) and now maintained by Yair Altman (2015-). When exporting to vector format (PDF or EPS) this function requires that ghostscript is installed on your system. Ghostscript can be downloaded from:

`http://www.ghostscript.com`.

When exporting images to eps and pdf formats, *export_fig* additionally requires pdftops, from the Xpdf suite of functions. This is included in the xpdf tools package and can be downloaded from:

`https://www.xpdfreader.com/download.html`

4
Loading images

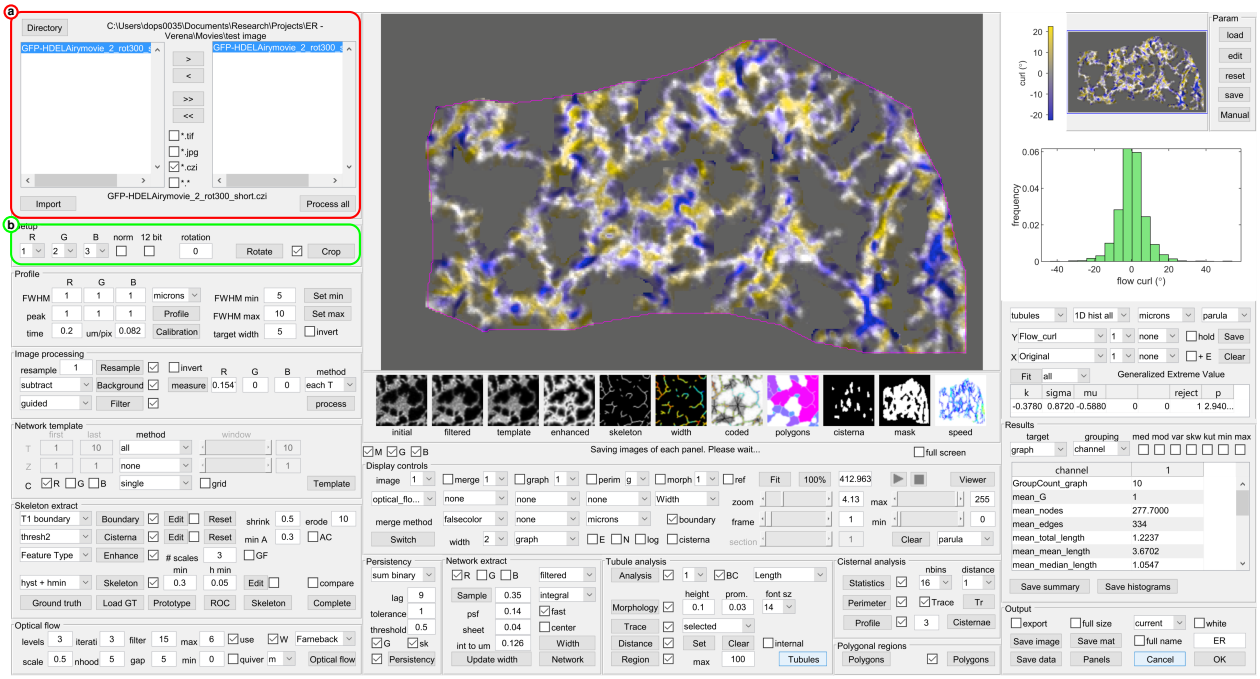


Figure 4.1: (a) The Image load panel, and (b) the Setup panel

4.1 The Image load panel

The ER network GUI automatically displays any image files present in the current directory with the default ***.tif** extension when the program is started (Fig. 4.2). The working directory can be changed using the **Directory** button, and additional file types can be displayed using the ***.jpg**, ***.png** or ***.*** checkboxes.

A single image can be selected for processing in the left-hand list box using a left mouse-click to highlight the name in the list followed by clicking the **>** arrow. Multiple files in any combination can be selected using *Ctrl + left click* to highlight the files, followed by the **>** arrow. Alternatively, all the files can be selected with the **>>** arrow and will appear in the right-hand list box. Individual files, or all the files can be removed using the **<** and **<<** arrows, respectively.

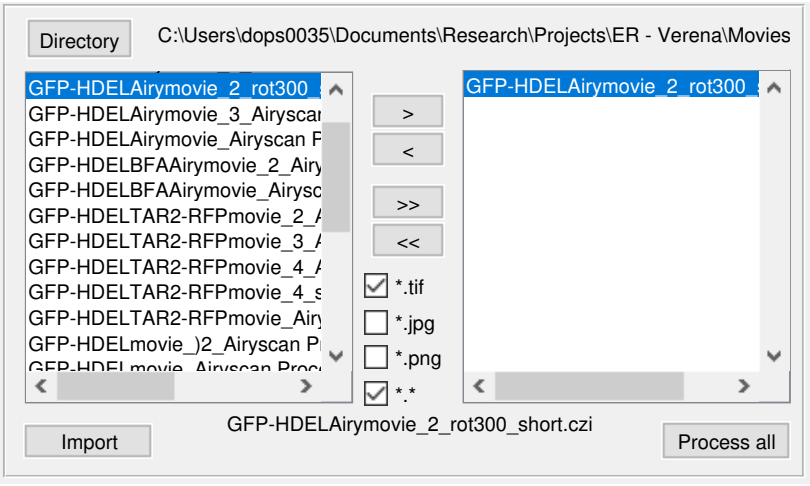


Figure 4.2: Image load panel: Images in a range of file formats can be imported for processing.

Many different filetypes can be opened directly using the Bio-Formats program[1] called by the import functions. Alternatively, greater control of the format of the imported image is available through the **Import** button in the **Setup** panel (Fig. 4.3). This opens up a new window to load, align, crop and filter the image, prior to network analysis, and is described in Chapter 13.

[1] M. Linkert, C. T. Rueden, C. Allan, J. M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. Macdonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. W. Eliceiri, and J. R. Swedlow. Metadata matters: access to image data in the real world. *J Cell Biol*, 189:777–82, 2010

4.2 *Defining the channel order and bit-depth*

In multi-channel images, the channel order may vary depending on the precise imaging setup. The **R,G,B** drop-down menus in the **Setup** panel can be used to re-order the channels to a standardised format. So, for example, it may be convenient to assign an HDEL or KDEL-marker delineating the ER lumen to channel 1, irrespective of whether it has been tagged with a red or green fluorophor.

Many image file formats store images at either 8-bit or 16-bit, whereas the data may be collected in 12-bit format. The **12-bit** check box ensures that the image is scaled to the full bit-depth. Once imported, images are normalised to the range 0-1. If the **norm** checkbox is ticked, each channel is normalised separately, otherwise normalisation is between the minimum and maximum of all channels.

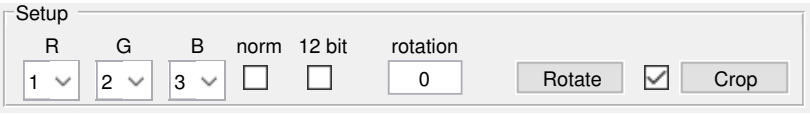


Figure 4.3: Setup panel: Allows the channel order to be changed and images normalised and cropped.

4.3 *Image rotation*

The image can be rotated by an arbitrary angle measured in degrees anticlockwise using the value in the **rotation** text box. The **Rotate**

button applies the rotation and displays the result on screen, scaled to fit the window. Regions outside the original image are set to zero.

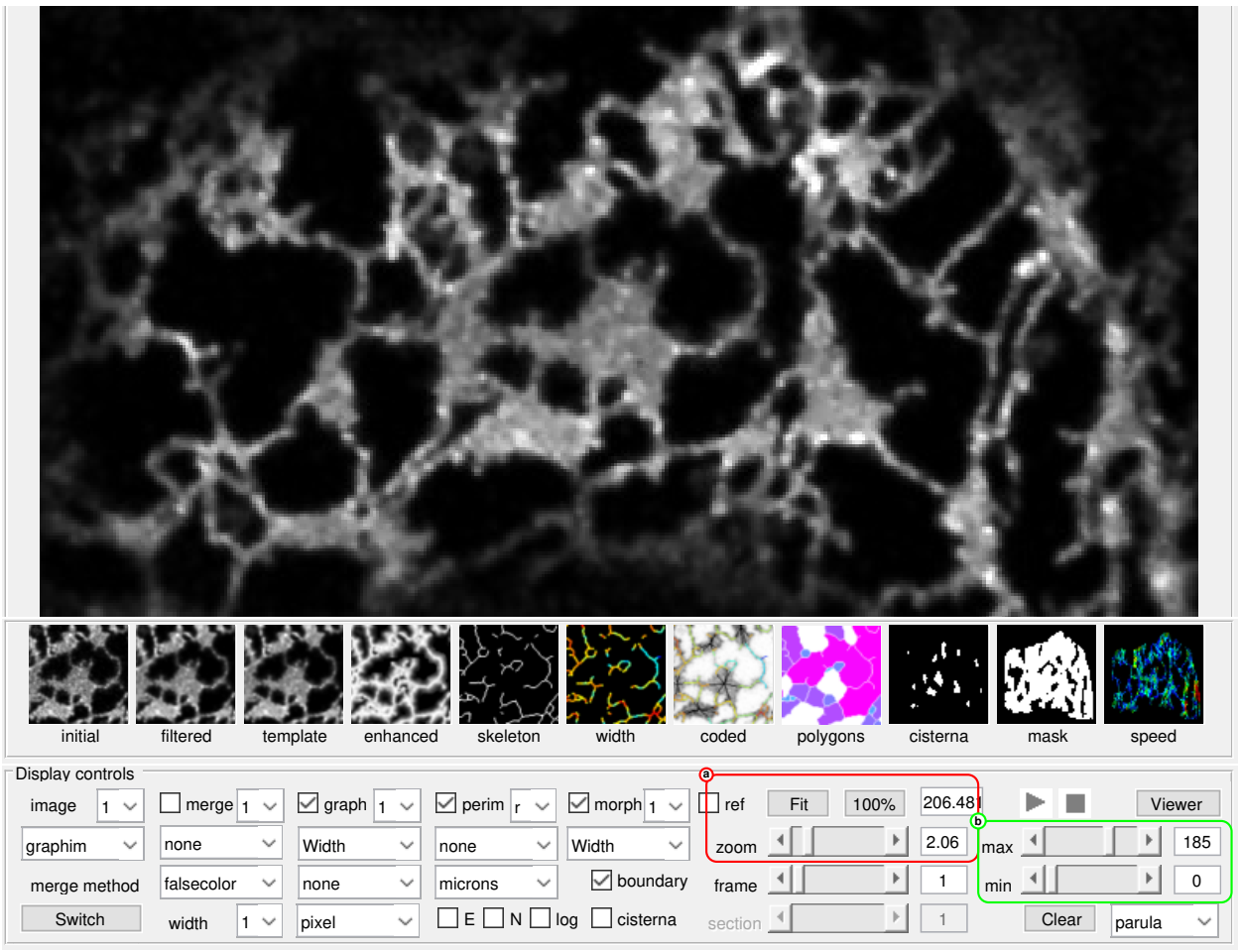
4.4 *Cropping the initial image*

The **crop** button will automatically show the original or rotated image, if applicable, sized to fill the display window, and prompts the user to drag a rectangle on the image display using the mouse to enclose the desired region-of-interest (ROI). On completion of the rectangle, the image is cropped and the resulting sub-region displayed. If there is an error in the region selected, the process can be repeated by clicking on the filename in the listbox to reload the original image. The co-ordinates of the cropped region are stored in a parameter file associated with the image and are re-applied every time the image is loaded if the adjacent **use** checkbox is ticked.

4.5 *Image display controls*

Once a filename is displayed in the right listbox, the image is automatically loaded and displayed in the main window (Fig. 4.4).

Figure 4.4: Main image display, thumbnail shortcut bar and associated controls to adjust zoom (a) and image contrast (b)



If multiple files have been selected, the last one in the sequence is shown. Any image can be viewed by clicking on it's filename in the right-hand listbox. At the same time, a thumbnail for the *initial* image is also displayed in the thumbnail shortcut bar immediately underneath the main display. Thumbnails for key steps in the processing sequence are displayed once the appropriate step has completed successfully, and can be used subsequently to switch rapidly between different images.

A number of options to adjust the image displayed are available in the **Display controls** panel. The main controls that are relevant at this stage adjust the image size and contrast, through the **zoom** (Fig. 4.4(a)), **white level** and **black level** sliders on the right of the panel (Fig. 4.4(b)). The **Fit** button resizes the image to ensure all of it is visible, whilst the **100%** button gives a 1:1 image:screen pixel scaling. The other commands in the **Display controls** panel will be covered at a later stage in the manual.

5
Profile measurements

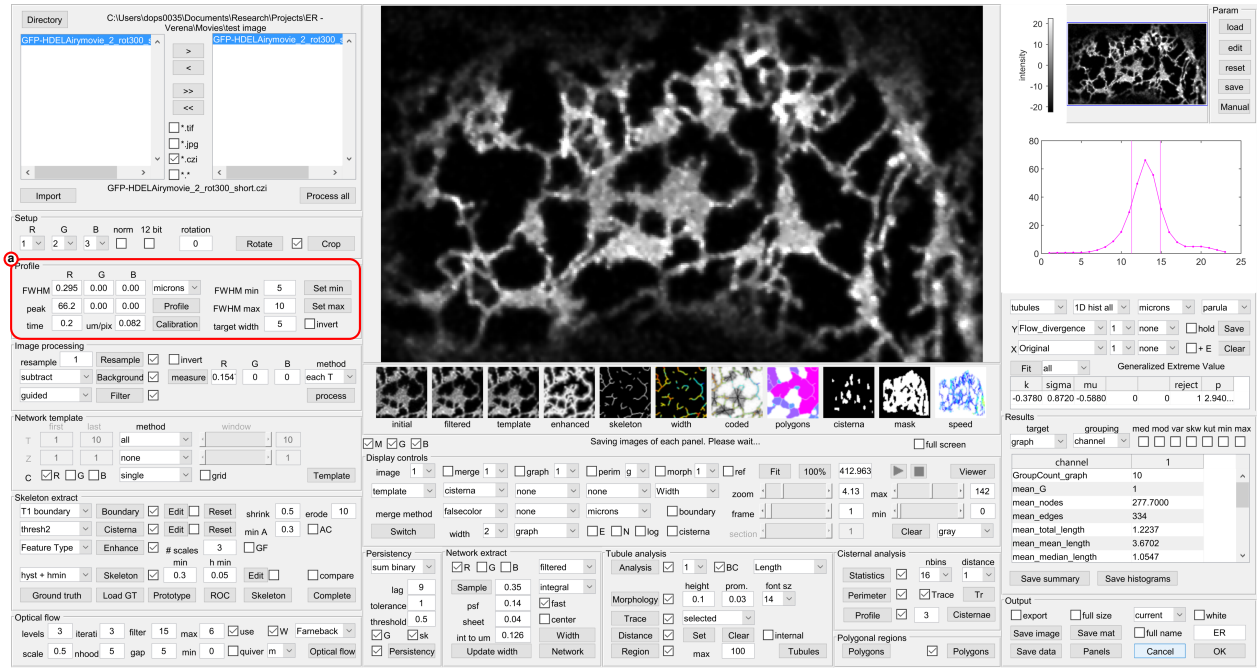


Figure 5.1: (a) The Profile panel

5.1 Measurement of the approximate tubule diameter

Images of the ER may have been collected at different pixel resolutions, depending on the microscope settings, and may span different width scales depending on the experimental treatment and genetic background. In addition, the pixel size may need to be reduced below the optimal Nyquist sampling value for the microscope to reduce downstream discretisation errors due to pixelation when approximating circular processing kernels.

To standardise all the subsequent processing steps, it is useful to define the expected minimum and maximum width of the tubular components manually using a transect drawn on the image, using the **Set min** and **Set max** buttons in the **Profile** panel (Fig. 5.2). The image can be subsequently resampled to ensure that the minimum tubule width is scaled to the width set in the **target** textbox.

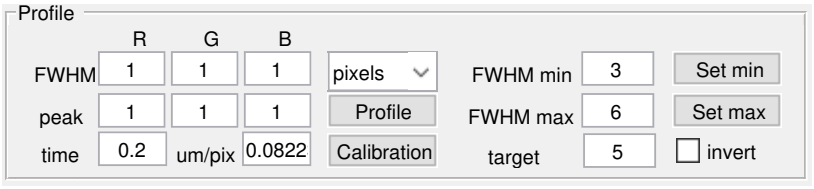


Figure 5.2: Image profile controls: these allow the user to measure the physical size of structures in the image from transects that automatically calculate the FWHM of the underlying feature. These are used to estimate the minimum and maximum diameters of the tubules to standardise all subsequent processing steps

When either the **Set min** or **Set max** buttons are clicked, the user is prompted to draw a two-point transect on the image across a tubule that, by eye, appears to be close to the smallest or largest tubule diameter, respectively. On completion of the second mouse click, a graph of the transect is displayed in the upper **graph** panel (Fig. 5.3), with the line colour reflecting the intensity values of the original RGB channels.

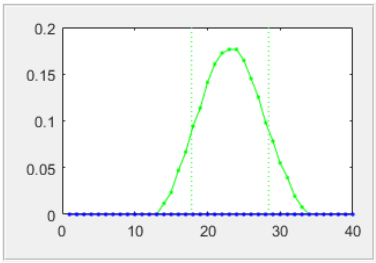


Figure 5.3: Profile measurements: the graph shows the intensity profile along a user-defined transect drawn on the image, along with the full-width half-maximum (FWHM) automatically calculated from the peak height, in this case for the green channel

In addition, the full-width at half-maximum (FWHM) peak intensity is automatically calculated and displayed as:

- two dotted vertical lines on the graph;
- the estimated pixel width in the **FWHM min** and **FWHM max** text boxes, respectively;
- the estimated pixel width and peak intensity (in normalised units) values in the **FWHM** and **peak** text boxes for the **R**,**G** and **B** channels;

Values for FWHM are given in pixels, whilst the peak intensities are given in normalised units ranging from 0 to 1. Additional profiles can be drawn at any stage without updating the **FWHM min** and **FWHM max** text boxes by using the **Profile** button.

The value of $FWHM_{min}$ is used to calculate a resampling factor needed to ensure that the minimum apparent tubule width matches the **target** width (typically 3-7 pixels wide, depending on the application) to reduce pixelation errors later on. Likewise, $FWHM_{max}$ is used to determine the number of scales to use in the subsequent steps to ensure both that the largest tubules are correctly segmented, and also that structures above this limit are identified as cisternae. As these scaling factors are approximate, it is sufficient to round the values of $FWHM_{min}$ and $FWHM_{max}$ to convenient integer values (usually odd numbers to ensure that the processing kernels are centered on the pixel of interest).

The **Calibration** button prompts the user to define the physical scale between two measurement points on the image, which then updates the adjacent textbox to give the pixel spacing in **micron per pixel**. Alternatively, the pixel size can be entered manually in the text box, if the information is available from the original image file. Likewise, the time interval between frames in time-series images can be entered in the **time** text box.

6

Image processing and extraction of a pixel skeleton

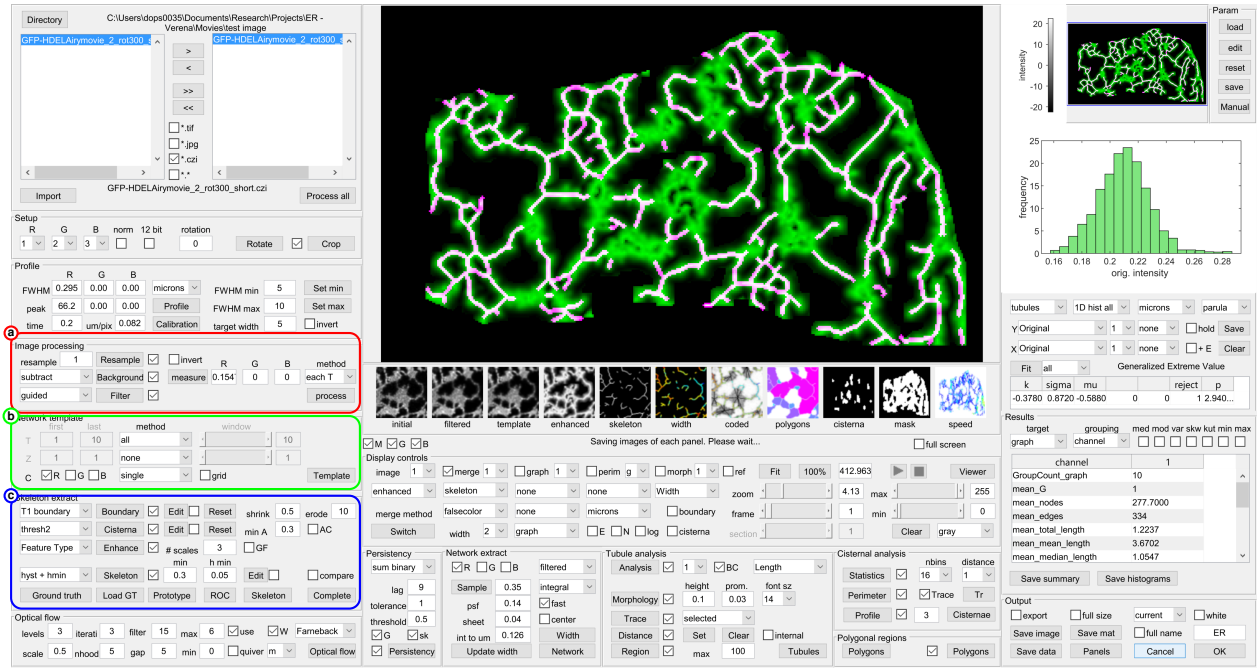


Figure 6.1: (a) The Image processing panel, (b) The Network template panel and (c) The Skeleton extract panel

6.1 General principles

Operations are grouped into a number of control panels including **Processing** (resampling, background subtraction and filtering), **Network template** construction, and the final **Skeleton extraction** (ridge enhancement, skeletonization). Each step has a number of options that can be selected from the adjacent dropdown menu on the left. In addition, some steps allow the user to manually modify the image produced at a particular step, or to set the values for specific parameters. The **use** checkboxes enable the user to toggle particular steps on or off to explore the impact on the final skeleton. If a particular step is not operational, the corresponding controls are greyed-out.

6.2 Image processing

The **Image processing** panel (Fig. 6.2) contains the controls needed to resample the image so that tubules are set to be a similar width (in pixels) for subsequent network extraction, subtract the background, and apply some level of smoothing.

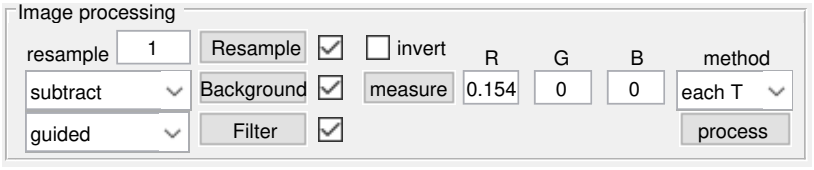


Figure 6.2: Image pre-processing panel: These controls set the parameters to resample, subtract background and smooth the image

Each step can be tested individually using the appropriate button. Alternatively the **process** button applies all the operations selected.

6.3 Image resampling

The value of $FWHM_{min}$ estimated from the **Profile** measurements is automatically used to calculate the resampling factor needed to ensure that the minimum tubule width matches the size set by the **target** text box. Likewise, the $FWHM_{max}$ value is used to determine the number of scales to use in the subsequent steps to ensure both that the largest tubules are correctly segmented, and also that structures above this limit are identified as cisternae. As the scaling required is approximate, it is sufficient to round the values of $FWHM_{min}$ and $FWHM_{max}$. The resample **use** checkbox is active by default. If the box is unchecked, no resampling takes place, but it is possible that subsequent steps do not perform as expected. Once the **resample** button is clicked, the initial image is resampled and displayed. The *resample* thumbnail is updated to show a small icon taken from the center of the resampled image.

6.4 Background measurement and correction

Accurate measurements of the fluorescence signal from the ER requires correction for instrument dark current, amplifier offset and background signal. The **Background** checkbox enables a number of options to automatically or manually correct the background contribution including:

- *'Subtract'* : This will subtract a constant value from the image and is appropriate for removing instrument black-level offsets or very diffuse fluorescence. The *value* is set in the adjacent text box and can be input manually, or measured using the **measure** button which prompts the user to define a background ROI on the image for the measurement.

- *'Opening'* : the image is processed with an opening function using a disk-shaped kernel with the radius set by $(1.2 \times FWHM_{max})/(2 \times resample)$ (i.e. 20% larger than the size of the largest tubule expected). This removes any features smaller than $1.2 \times FWHM_{max}$ and provides an estimate of the local background around each pixel. The opened image is subtracted from the original to correct for the local background. This method may be useful if there is some out-of-focus blur in the image or an amount of signal from another compartment, such as the cytoplasm. However, it is less useful if there are sheet-like regions larger than $1.2 \times FWHM_{max}$, as these remain after the opening operation and are then subtracted from the image, distorting the pixel intensities in the neighbouring regions.

- *'Surface fit'* : this finds all the local minima across the image and fits a surface to points in the 10-90% interval using a cubic polynomial. The surface is converted to an image and subtracted from the original. This method works if the network is sparse, giving plenty of local background estimates across the image.

- *'Sub low pass'* : the image is filtered using a Gaussian kernel with a large radius sufficient to remove all the high-frequency information in the image. The standard deviation for the Gaussian kernel is calculated as $2 \times FWHM_{max}$, or approximately twice the size of the largest tubule diameter. The low pass image is subtracted from the original and the image re-normalised. This approach also suffers if there are large sheet-like regions.

6.5 Filtering the image to improve signal-to-noise

Most simple noise reduction algorithms use isotropic kernels and smooth the image equally in all directions. This is not desirable when analysing the ER network, as the tubule boundaries will become blurred. Instead a number of adaptive anisotropic filters are provided that smooth within the tubular network structures, but do not spread across boundaries.

The **filter** controls can be used to locally smooth the image with various adaptive filters, and/or increase the contrast using contrast-limited adaptive histogram equalisation (CLAHE). The current options include:

- *'CLAHE'* : applies contrast-limited adaptive histogram equalisation[1] to the image to expand the contrast range over local regions.

- *'Coherence'* : applies an anisotropic diffusion filter written by Dirk-Jan Kroon[2] that smooths regions of low variance, but avoids blurring of object boundaries.

- *'Guided'* : applies an edge-preserving smoothing filter that is guided by the intensities in the original image[3]

[1] K. Zuiderveld. *Contrast limited adaptive histograph equalization*, pages 474–485. Graphic Gems IV. Academic Press Professional, San Diego:, 1994

[2] D. J. Kroon, C. H. Slump, and T. J. Maal. Optimized anisotropic rotational invariant diffusion scheme on cone-beam ct. *Med Image Comput Comput Assist Interv*, 13:221–8, 2010

[3] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1397–1409, 2013

- *'Coherence + CLAHE'* : applies the *'Coherence'* filter followed by histogram equalisation.
- *'Guided + CLAHE'* : applies the *'Guided'* filter followed by histogram equalisation.

The filtering and contrast adjustment are applied to aid segmentation of the pixel skeleton. Quantitative measurements of fluorescence intensities later on refer back to the background-corrected image intensities. CLAHE helps the user see detail across the whole image, but is only needed if the subsequent segmentation step is dependent on image intensities.

6.6 Template construction

The template image is constructed from different combinations of channels, sections and frames to ensure that the final pixel skeleton captures all the relevant information available on the ER structure. Thus, typically the ER-lumen is labelled with GFP-HDEL or RFP-HDEL, which provides a reasonable marker to segment the ER structure from a single channel on its own. If a second label is present, such as reticulon-RFP, for example, then the average or maximum intensity of both channels might provide a more complete image of the ER. The channels to combine are selected from the **R,G** and **B** check boxes, whilst the combination method is selected from the adjacent drop down menu.

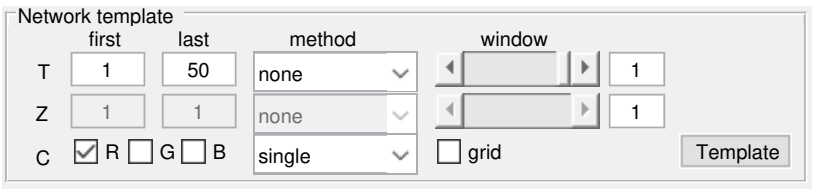


Figure 6.3: Setting up a template for ER network extraction

If the image is a time-series, a separate template can be defined for each image in the series, or from a limited range (set by the **first** and **last** text boxes), or as a combination of images over time, set by the **method** drop down menu. The **window** slider allows the selected combination method to be applied over a varying time window. This can be used to achieve a moving temporal average, for example. Nevertheless, whilst temporal averaging helps to remove noise, there is often too much movement in ER images for this to be useful.

6.7 Setting up a boundary mask

Once the template has been defined, it may be appropriate to define a boundary mask to exclude regions that should not be analysed, or to restrict the analysis to a specific region of the cell or tissue using the first set of controls in the **Skeleton extract** panel (Fig.

6.4(a)). The mask is calculated from the template image and has the same dimensionality. The default setting is to automatically apply a boundary, but un-ticking the boundary **use** checkbox will disable the boundary controls.

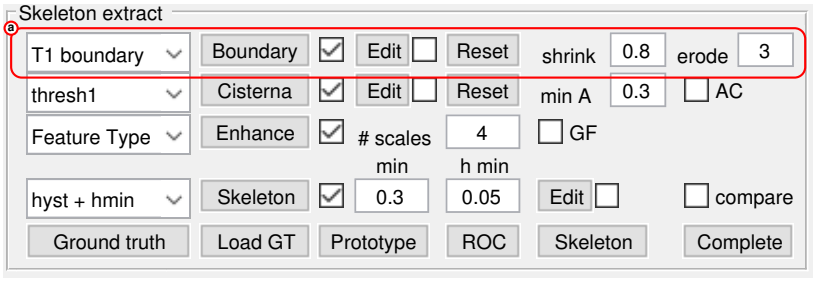


Figure 6.4: Controls for the boundary mask

Care is needed with this step as the thresholding operation may introduce breaks in some dim tubules, which are then not considered in subsequent processing steps. The adjacent dropdown menu provides a number of options including:

- *'Background'* : prompts the user to define a ROI in a background area of the image. The threshold is then calculated as the $background\ mean + 2 * SD\ units$.
- *'T1'* : this segments the main structures using an automatic threshold determined by Otsu's method[4] that minimizes the intraclass variance of the foreground and background distributions, but does not fill in any gaps or holes in the resulting binary image. Only the largest connected component is retained.
- *'T1 fill'* : fills in any holes in the *'T1'* mask. This helps to correct any small breaks in tubules within any fully enclosed polygonal region, but masks out fewer background regions from the subsequent enhancement and segmentation steps.
- *'T1 boundary'* : This initially segments the image using the *T1* threshold, but then calculates the convex hull of the segmented outline. This boundary is then collapsed down onto the most prominent concave sections by an amount set using the **Shrink** textbox (typically 0.5) and filled completely. This boundary can be eroded further using the value set by the **erode** textbox in pixels[5].
- *'T2'*, *'T2 fill'* and *'T2 boundary'* : these operate in a similar manner to the *'T1'* settings, but use the lowest value of a two-threshold partition of the image histogram and can be useful if there are a number of very bright structures that distort a single threshold separation of background and object. This can occur in the presence of fusiform ER bodies in *Arabidopsis*[6,7], for example.
- *'T3'*, *'T3 fill'* and *'T3 boundary'* : these operate in a similar manner, but use the lowest value of a three-threshold partition of the image histogram.

[4] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, Cyber.*, 9:62–66, 1979

[5] Note: The erode function can be applied to any of the other segmentation methods as well

[6] R.W. Ridge, Y. Uozumi, J. Plazinski, U.A. Hurley, and R.E. Williamson. Developmental transitions and dynamics of the cortical er of arabidopsis cells seen with green fluorescent protein. *Plant and Cell Physiology*, 40:1253–1261, 1999

[7] C. Hawes, C.and Saint-Jore, B. Martin, and H.-Q. Zheng. Er confirmed as the location of mystery organelles in arabidopsis plants expressing gfp! *Trends in Plant Science*, 6:245–246, 2001

- *'Holes'*, *'No holes'*, and *'erode'* : these are the same as the *'T1'* methods and are included for backward compatibility.
- *'manual'* : used if the mask is to be defined manually (see Chapter 15).

Clicking the **Boundary** button runs the chosen method, displays the segmented binary image alongside the initial grayscale image to aid comparison, and updates the boundary thumbnail (Fig. 6.5). The outermost boundary can be plotted if the **boundary** checkbox in the display controls panel is ticked. The area enclosed by the outermost boundary controls the estimate of the area covered by the network and impacts on any subsequent density estimates of tubules or cisternae.

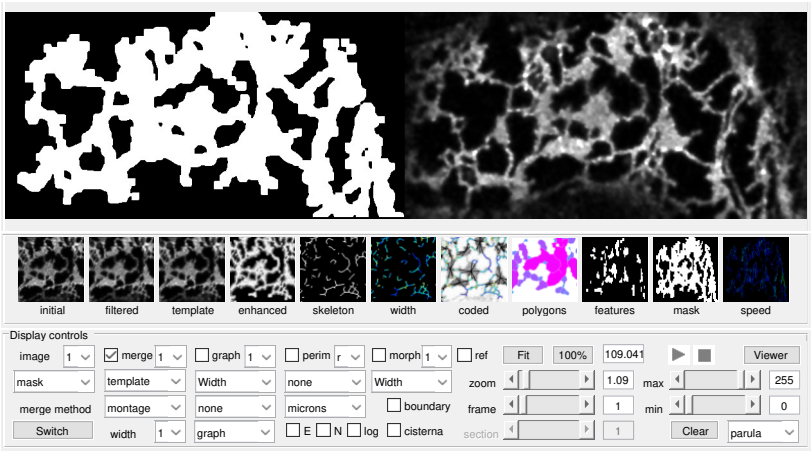


Figure 6.5: Side-by-side display of the boundary segmentation image using the 'T1' option. The boundary thumbnail is also updated

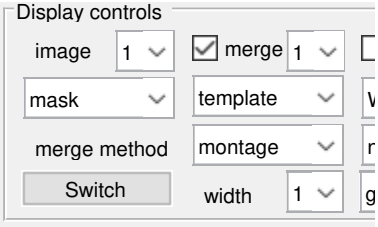


Figure 6.6: Display control settings for side-by-side montage

It is possible that the automatic boundary settings do not provide the desired masking of unwanted information, requiring the user to define the mask manually. The **edit** button will open an additional window with a set of tools to allow manual adjustment of the binary image. Full details of the binary editing program are given in Chapter 15. If a boundary image has been defined manually, the **use edit** check box is automatically activated and the manually defined boundary will be applied to subsequent processing steps. The manually-defined boundary is saved in the parameter file and can be re-applied or edited further anytime the analysis is re-run.

The **Reset** button removes the stored version of the manually-edited boundary mask. Use with care as this option may delete a lot of careful editing!

6.8 *Defining the ER cisternae*

In addition to the tubular-reticulum, the ER includes sheet-like regions called cisternae, or regions of closely appressed tubules that are difficult to separate[8]. The methods used to segment the tubular ER do not work with the cisternae, so a different set of

[8] C.J. Nixon-Abell, J.and Obara, A.V. Weigel, D. Li, W.R. Legant, C.S. Xu, H.A. Pasolli, K. Harvey, H.F. Hess, E. Betzig, C. Blackstone, and J. Lippincott-Schwartz. Increased spatiotemporal resolution reveals highly dynamic dense tubular matrices in the peripheral er. *Science*, 354:aaf3928, 2016

controls are available to allow the user to segment these structures independently (Fig. 6.7(b)).

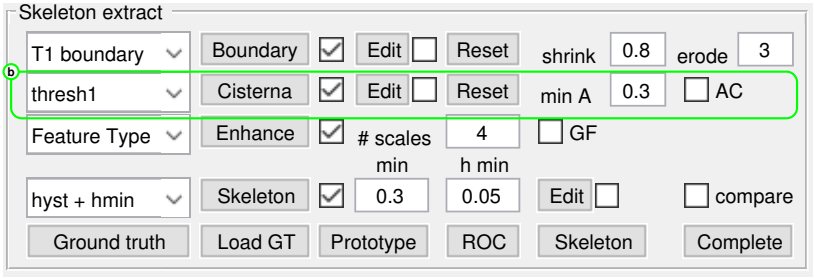


Figure 6.7: (b) Controls for segmentation of ER cisternae

To enable the **Cisterna** controls, the cisterna **use** check box needs to be ticked. The adjacent drop down menu then gives access to several different methods including:

- *'thresh1'* : Initially removes the tubular elements using greyscale image opening with a radius set to be 20% greater than $FWHM_{max}$ of the tubules, and converts the resultant image to a binary mask using Otsu's automatic threshold. This is effective at detecting larger sheet-like regions of the ER, but the resultant binary image tends to give blob like features that extend beyond the original cisternal regions.
- *'thresh2'*, *'thresh3'* : Initially removes the tubular elements using greyscale image opening with a radius set to be 20% greater than $FWHM_{max}$ of the tubules and then applies an automatic segmentation using the lower threshold of a two-threshold (*thresh2*) or three-threshold (*thresh3*) partitioning of the intensity histogram.
- *'Opening'* : operates as *'thresh1'* and is included for backward compatibility.

The **AC** checkbox applies an active-contour algorithm[9] to shrink the cisternae detected down to match the underlying intensity profile. This often performs better than a simple opening operation and is set as the default.

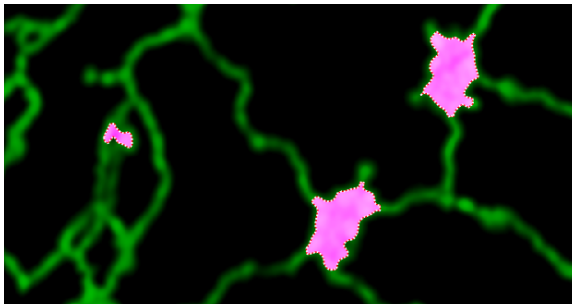
The initial cisternal segmentation is refined by an image opening operation with a radius of 2 pixels to break any thin connecting regions between adjacent cisternae, and the resulting objects are filtered on the basis of their area to be above 0.3 μm^2, chosen as the area of persistent *'puncta'* that are associated with immobile scaffold sites[10]. The minimum area (in μm^2) can be set using the **min** textbox. This ensures that puncta are included in as nodes in the tubular network, rather then being classified as small cisternae.

Clicking the **Cisterna** button runs the chosen method, updates the cisterna thumbnail, and displays the segmented binary image superimposed on the initial grayscale image in false-colour (Fig. 6.8 (a)). Once the cisternae have been defined, the **perim** checkbox

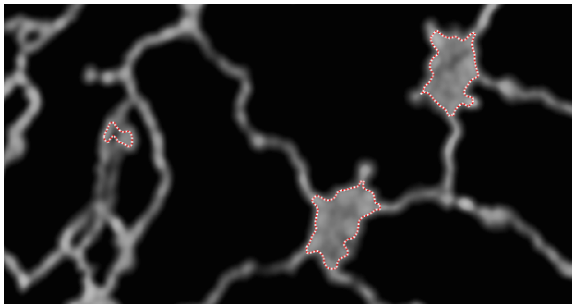
[9] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10:266–277, 2001

[10] I. Sparkes, J. Runions, C. Hawes, and L. Griffing. Movement and remodeling of the endoplasmic reticulum in nondividing cells of tobacco leaves. *The Plant Cell*, 21:3937–3949, 2009b

in the **display** panel will superimpose the perimeter outline on any displayed image in a dashed red border (Fig. 6.8 (b)), The colour can be changed using the dropdown menu. A number of morphology measurements of the cisternae are provided including the area, major and minor axis lengths, perimeter and solidity.



(a) Detection of ER cisternae and appressed tubues



(b) Feature boundaries

*Figure 6.8: Feature detection: (a) False-colour merge of the cisternae segmented using image opening and active contours that highlight ER cisternae or dense regions of appressed tubules. (b) The resulting cisternal boundary can be superimposed on any underlying image using the **perim** checkbox*

If no satisfactory segmentation of the cisternae can be achieved with the automatic settings, the cisterna **edit** button opens the binary editing window to allow manual adjustment of the binary image. Full details of the binary editing program are given in Chapter 15. If a cisternal image has been defined manually, the **use edit** checkbox is automatically activated and the manually defined cisternae will be used in subsequent processing steps. The edited cisternal image is saved with the parameter file and can be edited further or re-used any time the image is re-analysed. The **Reset** button deletes the edited image from the current analysis and the saved parameter file.

6.9 *Enhancing the tubular elements*

A number of options can be used to improve the relative contrast of tubular ER elements prior to segmentation, using kernels designed to pick-out 'ridge' like features, typically applied over a range of scales and angles (Fig. 6.9(c)).

The options include:

- *'PC, PCT and Feature Type'* : These use the phase-congruency method developed by Peter Kovesi[11],[12] which provides contrast-invariant ridge detection over a range of scales and angles. The MATLAB implementation[13] provides a number of outputs, in-

[11] P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1:1–26, 1999

[12] P. Kovesi. Phase congruency: A low-level image invariant. *Psychological Research*, 64:136–148., 2000a

[13] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing, 2000b. Available from: `http://www.peterkovesi.com/matlabfns/`

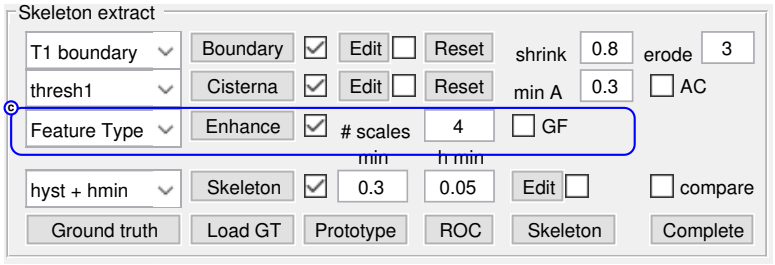


Figure 6.9: (c) Controls for tubule enhancement

cluding the level of phase congruency (*'PC' option*) as a measure of the edge strength (Fig. 6.10 (a)), the *Phase Congruency Tensor, PCT* that applies additional criteria to select ridge elements[14], and also the *Feature Type*, calculated as the weighted mean phase angle at every point in the image (Fig. 6.10 (b)). A value for the feature type of $pi/2$ corresponds to a bright line, 0 corresponds to a step and $-pi/2$ is a dark line. The feature type has proved to be one of the most robust and reliable outputs for subsequent segmentation[15], as all ridges, irrespective of their original intensity are identified with equal strength in the feature type image (Fig. 6.10 (b)).

- *'Vesselness'* : This calls the Matlab implementation of the classic Frangi[16] 'Vesselness' filter written by Marc Schrijver and Dirk-Jan Kroon and available from the Mathworks website. This gives a strong response for bright features where the second-order derivative of the image (Hessian) shows a strong anisotropy, but fails at junctions between tubules (Fig. 6.10 (c)).

- *'Neuriteness'* : Applies the second-order anisotropic Gaussian kernel originally proposed by Meijering *et al.* (2004)[17], which uses a slightly flattened second-order Gaussian kernel at a range of scales and angles to give better discrimination of ridge-like structures.

- *'SOAGK'* : Applies the multi-scale ridge detector developed by Shui *et al.* (2012)[18] and Lopez-Molina *et al.* (2015)[19] that uses second-order anisotropic Gaussian kernels (SOAGK). These can be configured flexibly in terms of size, orientation and anisotropy. This gives good ridge enhancement, but still retains the variation in local intensity along the tubules that can make subsequent segmentation more difficult (Fig. 6.10 (d)).

More details of the operation of each enhancement method can be found in Chapter 2.

The **GF** check box applies a subsequent *'Guided'* filter to smooth the result of the enhancement step. This can help to connect tubules at junctions where several of the enhancement algorithms tend to leave gaps (e.g. Fig. 6.10(c)). For subsequent processing the enhanced image is normalised to the range [0 1].

In each case the number of scales is initially set by the # *scales* parameter, determined from the ratio of $FWHM_{max}/FWHM_{min}$,

[14] B. Obara, V. Grau, and M. D. Fricker. A bioimage informatics approach to automatically extract complex fungal networks. *Bioinformatics*, 28:2374–81, 2012

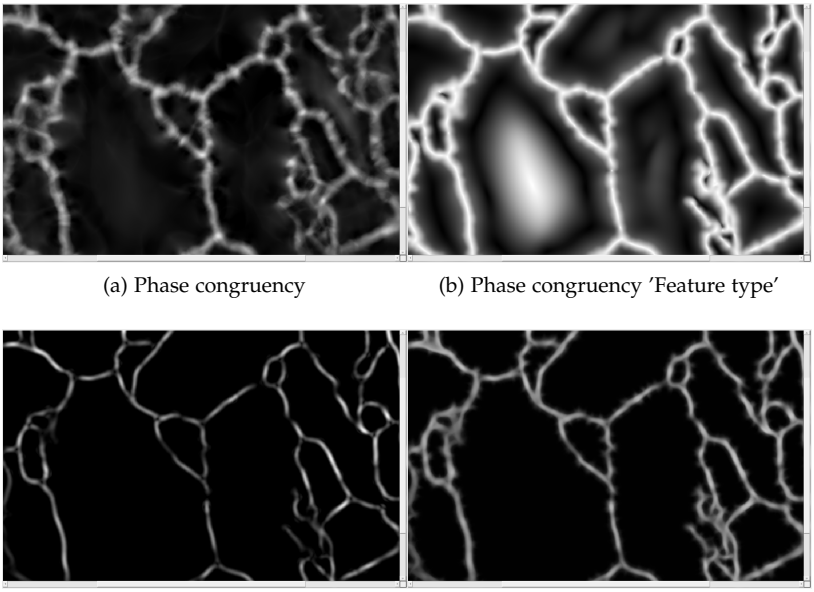
[15] M.D. Fricker, D. Akita, L.L.M. Heaton, N. Jones, B. Obara, and T. Nakagaki. Automated analysis of physarum network structure and dynamics. *J. Phys. D*, 50:254005, 2017

[16] A. F. Frangi, W. J. Niessen, K.L. Vincken, and M.A. Viergever. *Multiscale vessel enhancement filtering*, pages 130–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998

[17] E. Meijering, M. Jacob, J. Sarria, P. Steiner, H. Hirling, and M Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry*, 58: 167 – 176, 2004

[18] P.-L. Shui and W.-C. Zhang. Noise-robust edge detector combining isotropic and anisotropic gaussian kernels. *Pattern Recognition*, 45:806 – 820, 2012

[19] C. Lopez-Molina, G. V. D. de Ulzurrun, J. M. Baetens, J. Van den Bulcke, and B. De Baets. Unsupervised ridge detection using second order anisotropic gaussian kernels. *Signal Processing*, 116:55–67, 2015



(a) Phase congruency (b) Phase congruency 'Feature type'

(c) 'Vesselness' filter (d) Second-order anisotropic Gaussian kernel (SOAGK)

Figure 6.10: Comparison of different multi-scale approaches to ridge enhancement. (a) the raw output of the phase congruency as an intensity-independent measure of ridge strength (Kovesi, 1999,2000a,b); (b) the 'Feature Type' output of the phase congruency method. Note the strong edge response irrespective of the original tubule intensity; (c) the classic Frangi et al. (1998) 'Vesselness' filter based on the asymmetry of the isotropic second-order image gradient (note the 'gaps' that appear at the tubule junctions); (d) the output of second-order anisotropic Gaussian kernels (SOAGK) according to Lopez-Molina et al. (2015)

with a minimum bound of 3. However, this can be over-ridden using the adjacent textbox. This may be necessary to ensure that larger features, such as cisternae, are also processed prior to segmentation. The number of orientations is fixed at 6. In the case of the anisotropic Lopez-Molina filters, the anisotropy is set at 1.3.

6.10 Editing hidden parameters

The most useful parameters are typically accessible directly from the GUI interface. However, some of the algorithms take a number of arguments that are not readily visible. These values can be edited using the **Edit** button in the **Param** panel (Fig. 6.11), which opens up a table showing the default setting and the current choice for all the underlying parameters (Fig. 6.12). If the parameter selected has a number of different options set by a dropdown menu, these are displayed in the **options** box alongside. The value of the parameter can be changed by typing the new value into the current column, or selecting the option from the list box.

Figure 6.11: Controls to load, edit and save all the parameters

6.11 Skeletonization

The aim of the skeletonization step is to convert the enhanced image to a one-pixel wide skeleton along the centre-line of the tubule ridges, and is one area where up-sampling the image during the rescale operation may be beneficial, but at the expense of computational time. This is only an approximation to the true centre-line due to the pixel discretisation errors. There are two main approaches that can be used. The relevant controls are shown in Fig 6.13.

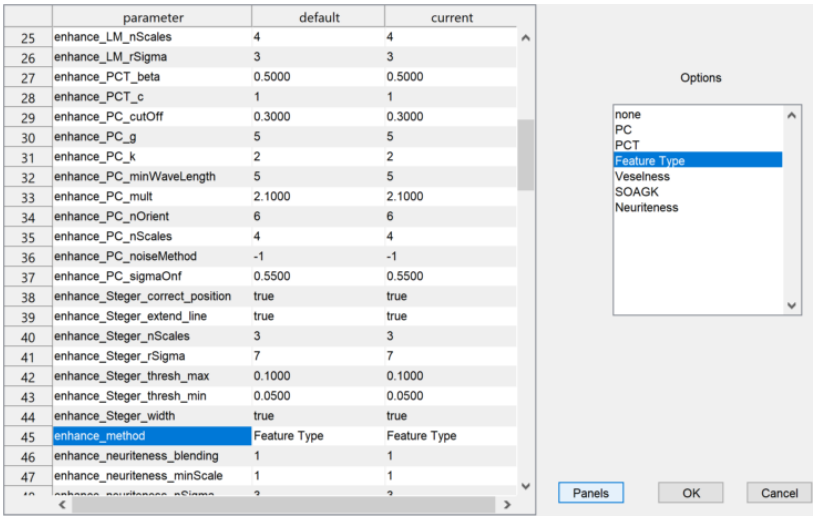


Figure 6.12: Interface to edit any hidden parameters

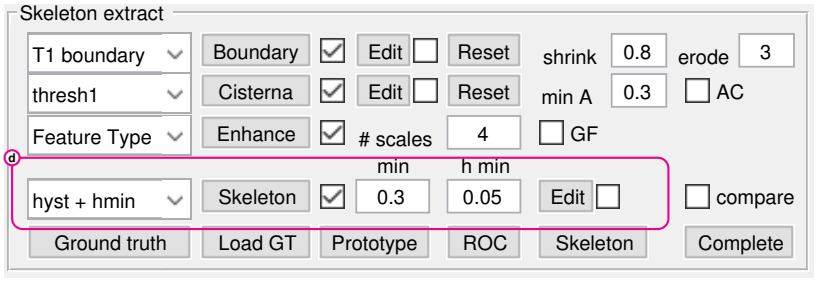
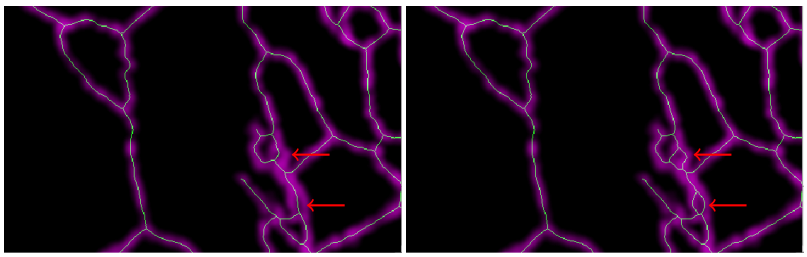


Figure 6.13: (d) Controls for skeletonization

The first method is 'hysteresis thresholding' (Fig. 6.14 (a,b)), which uses intensity information and some degree of pixel connectivity to provide an initial binary image that is then thinned to give a single pixel skeleton. The second is 'watershed thresholding' (Fig. 6.14 (c,d)) which follows connected ridges, irrespective of the absolute intensity, and automatically generates a single pixel skeleton.

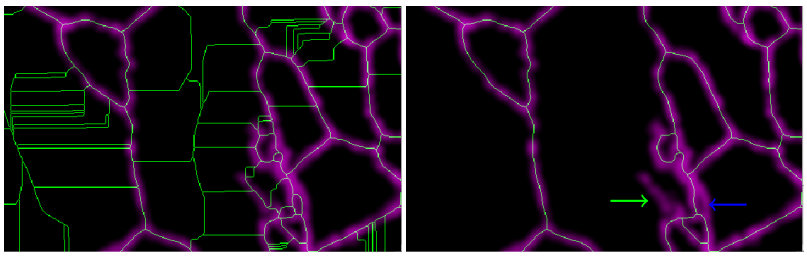
Hysteresis thresholding starts with seed pixels above the upper threshold, and then propagates the initial segmentation as long as pixels remain above the lower threshold. The resulting binary image is then thinned to give the single-pixel skeleton. The value of the lower threshold is critical - too high and the network becomes disconnected; too low and large blocks of the image are included in the resultant binary image that may fuse separate tubules into a single object. When this block is thinned, the skeleton does not map onto the ridge centre-lines. In addition, as the thinning process is not guided by the intensities in the enhanced image, the skeleton does not necessarily converge on the expected pattern at junction points (Fig. 6.14 (a,b)).

The watershed is better at segmenting the centre-line of the ridges and can handle variations in intensity well. However, it does not include any tubules that have a free end, and has a tendency to over-segment regions with noise (e.g. Fig. 6.14 (c)). Over-



(a) Overlay of pixel skeleton using hysteresis thesholding

(b) Improvement in the pixel-skeleton with inclusion of the h-minimum transform



(c) Over-segmentation using an unconstrained watershed segmentation

(d) Pixel skeleton following a constrained watershed segmentation

Figure 6.14: Comparison of different approaches to extraction of the pixel skeleton. (a) segmentation using hysteresis thresholding. Note that some adjacent regions are not separated and the skeleton crosses the void between the tubules (red arrows); (b) Improvement in segmentation following an h-minimum transform and imposing a local minimum. The adjacent tubules are now correctly segmented; (c) segmentation using an unconstrained watershed. This over-segments regions of background, that still have almost invisible ridge lines in the noise; (d) suppression of watershed over-segmentation using the h-minimum transform and imposed local minima. Note that the free tubule (green arrow) is lost in the watershed segmentation, and some arms of the loops are lost (blue arrow)

segmentation can be avoided using a boundary mask (with no fill), and/or by including additional steps that suppress regions with small intensity fluctuations, such as the h-minimum transform (Fig. 6.14 (d)).

These various options are selected from the drop down menu as follows:

- *'hysteresis'* : applies hysteresis thresholding using a lower threshold set by the adjacent textbox. The upper threshold used to define the seed points is automatically set as *lower threshold* + 0.2 6.14 (b)).
- *'watershed'* : applies a watershed segmentation and then extracts the watershed lines as the pixel skeleton. 6.14 (c)).
- *'hist + hmin'* : applies an h-minimum transform to smooth out regions with low fluctuations in intensity, and then sets these regions as local minima to ensure that the surrounding ridges will be segmented individually and do not spread into the basin, even if the absolute values are above the lower hysteresis threshold 6.14 (b)).
- *'WS + hmin'* : uses the same h-minimum transform and imposes local minima before the watershed operation, to prevent over-segmentation of irrelevant ridges arising from noise within the basins 6.14 (d)).

6.12 *Modifying the skeleton to accommodate ER cisternae*

The initial pixel skeleton will include both tubular and cisternal regions, even though a 'skeleton' is not necessarily a good repre-

sentation of the cisternal morphology. If the cisternae have been segmented earlier in the sequence, these regions are punched out of the pixel skeleton so that they do not contribute to analysis of the tubular elements. During the network extraction steps, described in (Chapter 9), the cisternal regions are represented as a set of linear connections radiating out from the center to connect with the tubules incident on the boundary. This ensures the overall connectivity of the network is retained.

6.13 *Manual editing of the pixel skeleton*

If the automated methods to delineate the pixel skeleton are still unsatisfactory, the skeleton can be edited manually using the **edit** button. This opens the binary editing program described in Chapter 15. The manually-edited skeleton is saved in the parameters file and can be re-applied if the **use edit** checkbox is ticked.

6.14 *Comparison with a manually-defined ground-truth*

The extent that the automatically extracted pixel skeleton captures the underlying structure of the ER is difficult to evaluate objectively. At the simplest level, the skeleton can be overlaid on the initial image for visual comparison (Fig. 6.16 (a,c)).

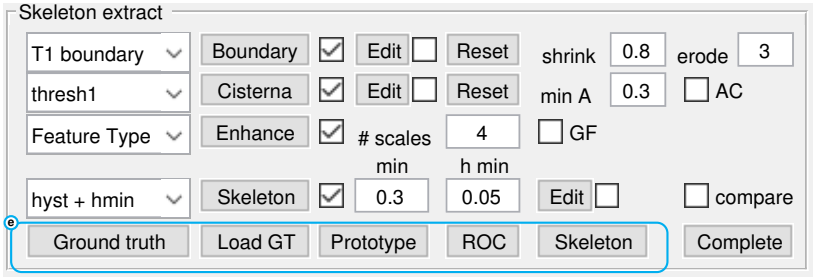
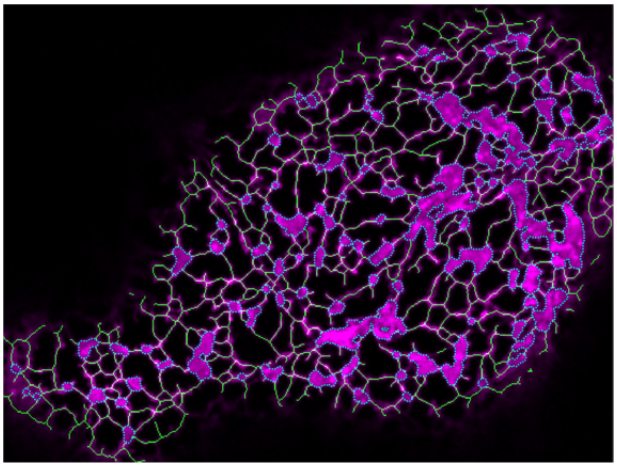


Figure 6.15: (e) Controls for comparison with a manually-defined ground truth

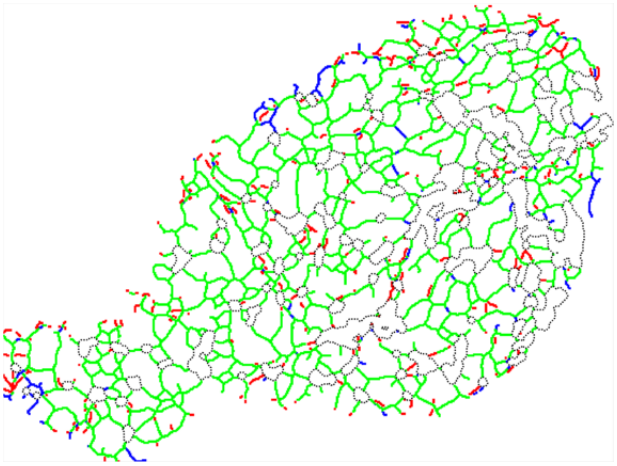
A more robust approach is to compare the extracted skeleton with a manually-defined skeleton, representing a ground-truth reference as judged by an expert human observer. The simplest method is to manually digitise the network using a digitising tablet, superimposed on the original image at the same scale. The digitised ground-truth image is loaded using the **Load GT** button (Fig. 6.15(e)), and processed through the same resampling, cropping and masking routines as the actual image, to ensure correct alignment. The degree of overlap between the automatic and ground-truth skeletons can be assessed visually in the merged image (Fig. 6.16 (d)).

Results can be quantified using the **ROC** (Receiver-Operating Characteristic) button, by counting the number of matched points ('True positives', TPs), within some tolerance, typically set to be half the tubule width, un-matched points ('false negatives', FNs),

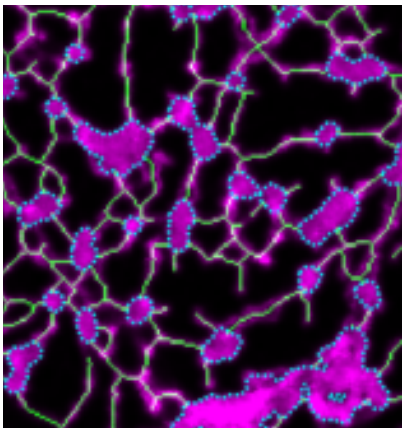
Figure 6.16: Comparison of performance against a manual ground-truth



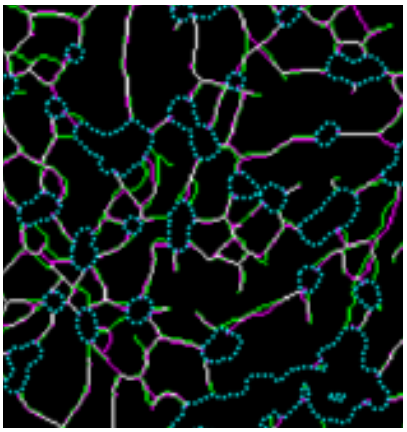
(a) Skeleton (green) overlaid on original image (magenta). Dotted cyan lines indicate the cisternae



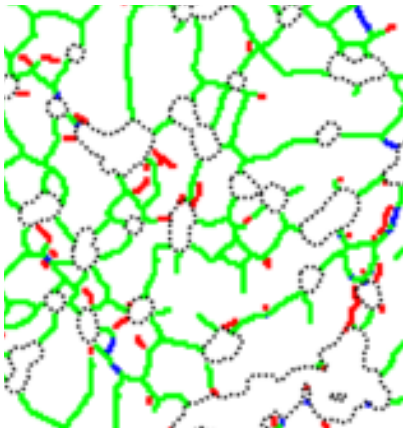
(b) Output of ROC analysis. Dotted black lines indicate the cisternae



(c) Zoomed region of skeleton (green) overlaid on original image (magenta)



(d) Zoomed region of skeleton overlay



(e) Merge between automatic skeleton (green) and ground-truth (magenta)

spurious points ('False positives', FPs). True Negatives' (TNs) are ignored as there are far too many TN background pixels. The results can be colour-coded to aid visualisation, with TPs labelled in green, FPs in blue and FNs in red (Fig. 6.16 (b,e)).

6.15 *Optimising parameter settings using Precision-Recall analysis*

It is also possible to use the ground-truth comparison as a means to optimise the enhancement and segmentation parameters. In essence, a small subset of the image is used as a test case to systematically vary each parameter in turn, in a sensitivity analysis. Clicking the **prototype** button, prompts the user to select a region of the image for comparison, and then opens a separate *Parameter selection* window (Fig. 6.17).

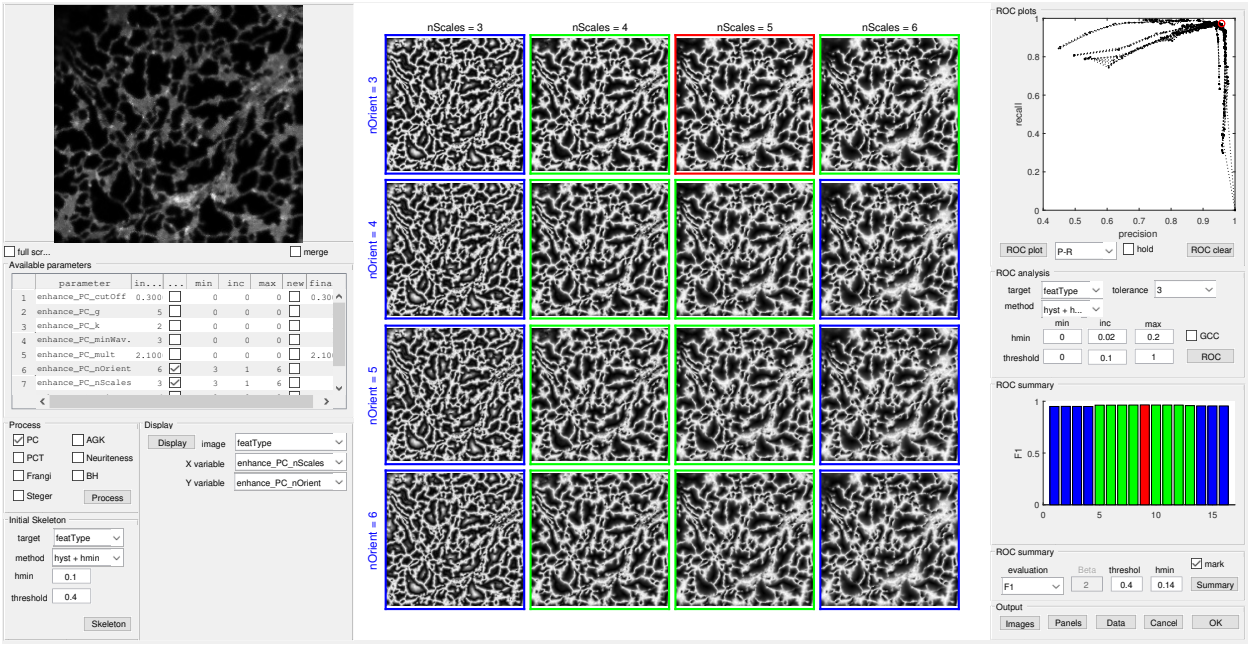


Figure 6.17: GUI interface for optimising parameter settings using Precision-Recall analysis

The full set of parameters available for the enhancement and skeletonisation methods are displayed in a table and a range of values for a sub-set of parameters can be selected. The segmentation is then run for a factorial combination of these parameters. Results are presented in a grid format for two (user-selected) parameters at a time to aid visualisation. Quantitative estimates of the best parameters use Receiver-Operating Characteristic (ROC) or Precision-Recall (P-R) analysis to find the best trade-off in parameter values that maximises the true hit rate (TPs), without increasing FPs and FNs too much. Full details are given in Chapter 16.

7
Estimation of ER velocity using optical flow

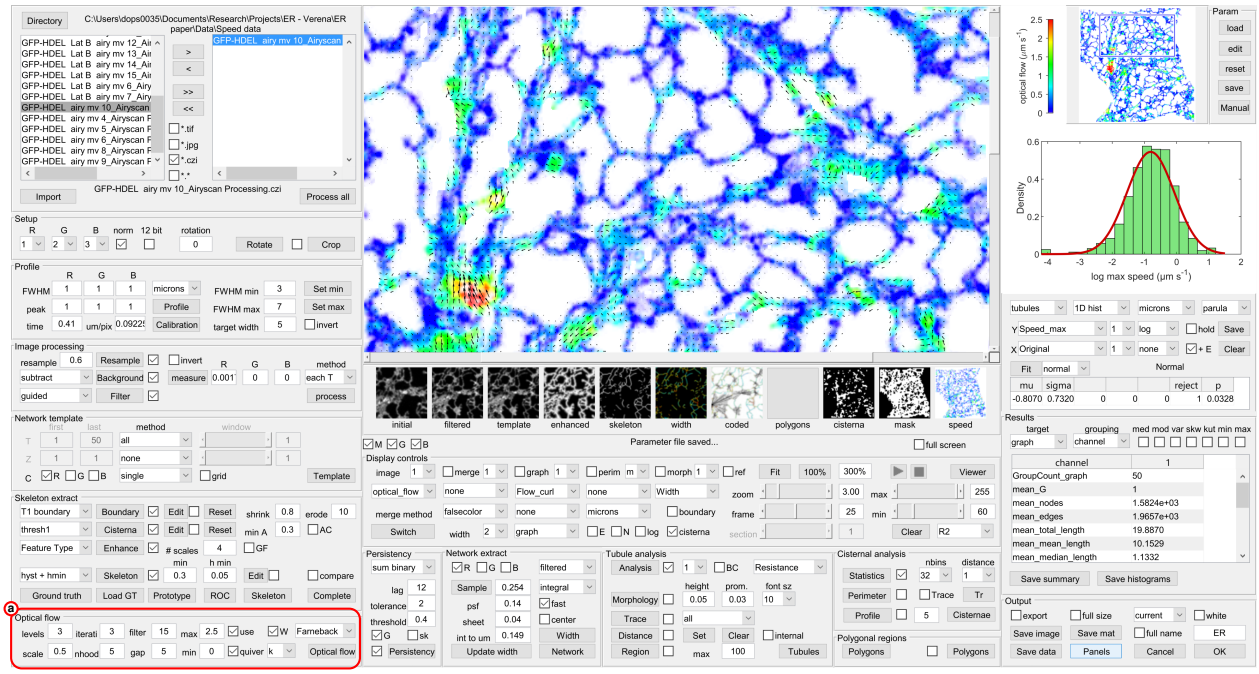


Figure 7.1: Optical flow controls

7.1 Introduction

The ER in plants is highly dynamic, with particularly rapid movement along actin bundles in cytoplasmic streams in the sub-cortical cytoplasm and trans-vacuolar strands. The first quantitative measurements of ER movement used an ImageJ plugin called KbiFlow (later repackaged as Lpx Flow)[1]. Typically 100 frames at 50ms intervals were analysed by initially removing static components by subtraction of the time-averaged image, then measuring the velocity using spatial-temporal correlation analysis. Additional masking was used to remove noisy parts of the data, such as the vacuoles. Using this approach, the maximum velocity recorded in $2.54 \times 2.54\,\mu m$ of the cortical cytoplasm was $1.35\,\mu m\,s^{-1}$ and $2.2\,\mu m\,s^{-1}$ deeper within the cell. Stefano *et al* (2014)[2] showed that motility of the ER increased during cell expansion from around

[1] H. Ueda, E. Yokota, N. Kutsuna, T. Shimada, K. Tamura, T. Shimmen, S. Hasezawa, V. V. Dolja, and I. Hara-Nishimura. Myosin-dependent endoplasmic reticulum motility and f-actin organization in plant cells. *Proc Natl Acad Sci U S A*, 107:6894–9, 2010

[2] G. Stefano, L. Renna, and F. Brandizzi. The endoplasmic reticulum exerts control over organelle streaming during cell expansion. *J Cell Sci*, 127: 947–53, 2014

$0.5\,\mu m\,s^{-1}$ at 3 days after germination (DAG) to $1.5\,\mu m\,s^{-1}$ at 12 DAG. Speeds in the thicker cytoplasmic strands running deeper into the cell were significantly faster, the maximum velocities around $4.2\,\mu m\,s^{-1}$ in 7-day old *Arabidopsis* cotyledons[3].

[3] K. Ueda, H.and Tamura and I. Hara-Nishimura. Functions of plant-specific myosin xi: from intracellular motility to plant postures. *Curr. Op. Plant Biol.*, 28:30–38, 2015

7.2 *Approaches to velocity measurement*

Horn and Schunck (1981) introduced a measurement of optical flow[4], based on local image gradients, coupled with the assumption that there are structures at a local scale within the image that translate in *x,y* over short time periods such that the pixel intensities of each object do not change between consecutive frames. This led to the image gradient or optical flow constraint equation:

[4] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17:185–203, 1981

$$I_x u + I_y v + I_t = 0 \tag{7.1}$$

Where I_x and I_y are spatial image gradients, I_t is the temporal image gradient, u is the horizontal optical flow, and v is the vertical optical flow.

To provide sufficient constraints to solve for u and v, it is also assumed that movement of neighbouring pixels forming the object will have similar velocity (i.e. flows are locally smooth) .

An alternative measure of optical flow that was originally formulated as an image registration technique by Lucas and Kanade (1981)[5], also used intensity gradient information to direct the search for the best match between two images. This scheme used a local approximation of the image intensity gradient, and then used an iterative least squares minimisation to find the displacement that minimised the difference between the two curves. The solution was further improved by including a weighting function favouring regions that match the linear approximation well.

[5] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 674–679, 1981

Rather than compute image gradients, Farnebäck[6] used local quadratic polynomial expansions to approximate local image intensities over a given neighbourhood, with weightings based on the strength of the signal and the distance from the central pixel. The algorithm also included a coarse-to-fine pyramidal scale-space iteration to improve the a priori estimate of the initial displacement field, and also handle larger displacements.

[6] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370, 2003

7.3 *Matlab implementations*

These three different optical flow methods are provided within the MATLAB Computer Vision Toolbox and called by the GUI, along with a variant of the Lucas-Kanade algorithm that includes a Difference-of-Gaussian (DoG) filtering step[7]. Each method has a number of tuneable parameters that are automatically displayed or greyed-out when the main method is selected.

[7] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994

7.3.1 *Farneback method*

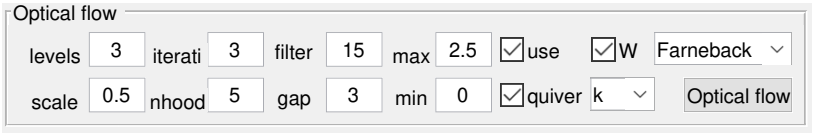


Figure 7.2: Controls to estimate local velocity in time-series images using the Farneback method

The Farneback method[8] (Fig. 7.2) fits a polynomial expansion to the neighbourhood around each pixel (given by the *nhood* size), and calculates the displacement from the quadratic coefficients, smoothed by a Gaussian filter (given by *filter size*). Typically three pyramid *levels*, with a *scale* of 0.5 gives sampling over four times the initial neighbourhood size to capture fast-moving objects.

[8] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370, 2003

7.3.2 *Horn-Schunck method*

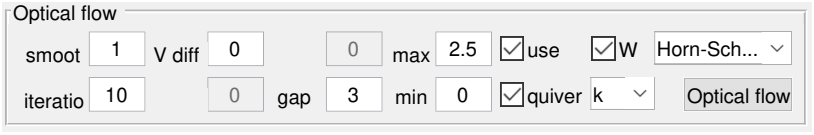


Figure 7.3: Controls to estimate local velocity in time-series images using the Horn-Schunck method

The Horn-Schunck method[9] (Fig. 7.3) calculates the local image gradients in x, y using a Sobel edge filter and a simple two-frame differencing for t. The solution is further constrained by a global smoothness term. Increasing the *smoothness* compensates for faster motion, whilst increasing the number of *iterations*, or decreasing the minimum velocity difference (*V diff*) helps to track slow moving regions.

[9] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17:185–203, 1981

7.3.3 *Lucas-Kanade method*

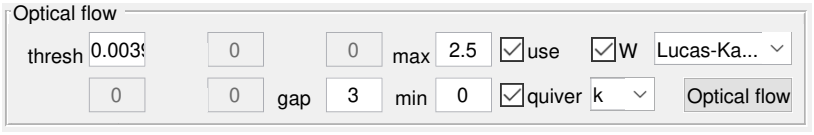


Figure 7.4: Controls to estimate local velocity in time-series images using the Lucs-Kanade method

The Lucas-Kanade method[10] (Fig. 7.4) calculates local image gradients in x, y and t using a set of fixed kernels with additional *smoothing*, before solving for u and v. Eigenvalues below the noise *threshold* are treated as zero velocity.

[10] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 674–679, 1981

7.3.4 *Lucas-Kanade Difference of Gaussians (DoG) method*

The Lucas-Kanade Difference of Gaussians (DoG) method[11] (Fig. 7.5) applies Gaussian smoothing in x,y (set by *sigma1*) and t (set by *frames*), before calculating the image derivatives. The derivatives are

[11] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994

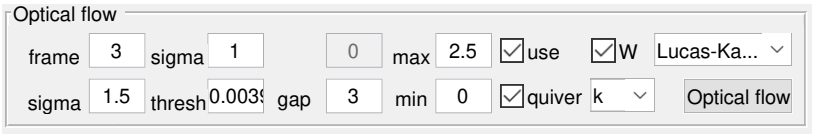


Figure 7.5: Controls to estimate local velocity in time-series images using the Lucas-Kanade DoG method

smoothed using a Gaussian filter (*sigma2*), before solving for u and v. Eigenvalues below the noise *threshold* are treated as zero velocity.

7.4 Quantitative results for Optical Flow

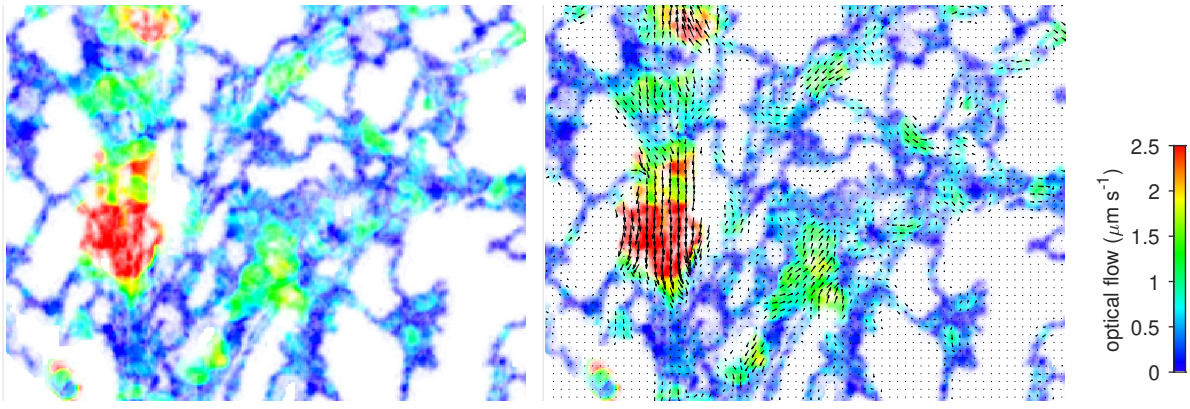
The Farneback algorithm[12], operating over three or four pyramid scales, appears to give the most robust results for plant ER and is mapped onto a colour-coded representation of the speed of movement (Fig. 7.6). This is probably because it is the only algorithm implemented here that operates over several pyramid scales, and is therefore able to capture the different rates of movement observed, particularly for fast streaming. The background is set to white if the **W** checkbox is ticked.

[12] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370, 2003

Kimura *et al.* (2017)[13] report similar advantages using the Farneback algorithm to analyse organelle movement in *Caenorhabditis elegans* embryos. The other algorithms capture movement of small bulges along the tubules and cisternae, but tend to miss larger-scale movements.

[13] K. Kimura, A. Mamane, T. Sasaki, K. Sato, J. Takagi, R. Niwayama, Y. Hufnagel, L.and Shimamoto, J.-F. Joanny, S. Uchida, and A. Kimura. Endoplasmic-reticulum-mediated microtubule alignment governs cytoplasmic streaming. *Nat Cell Biol*, 19: 399–406, 2017

The vectors describing the local flow can also be superimposed on the velocity image (or indeed any image displayed) using the **quiver** checkbox in the **Optical flow** panel (Fig. 7.6(a)). The spacing of the vectors is controlled by the **spacing** textbox, which also affects the relative scaling of the quiver magnitude. The **Colour** of the quiver arrows can also be selected from the dropdown menu.

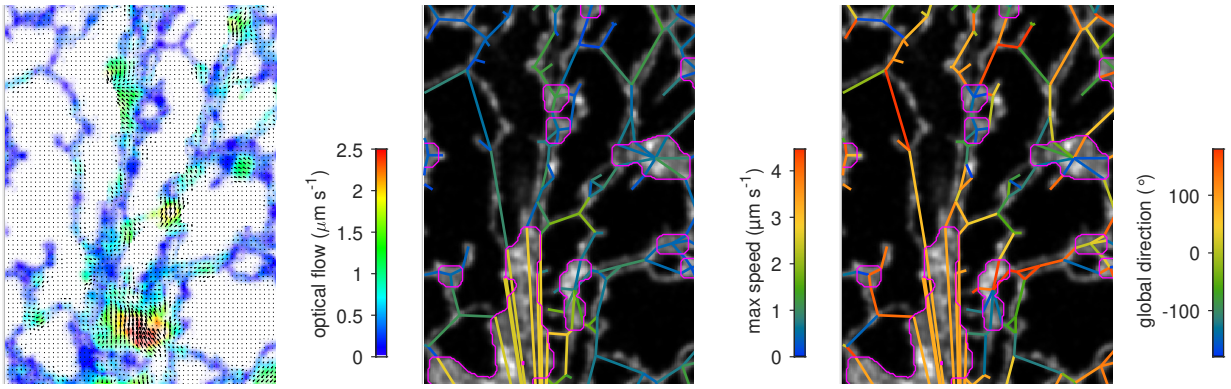


(a) Colour-coded optical flow (b) Quiver plot of velocity

Figure 7.6: Colour-coded representation and quiver plot of the local ER speed determined by the Farneback method

The colour-coded image provides some indication of the speed of movement by reference to the calibrated colour-bar, with the scale set between **min** and **max**. Full quantitative measurements are not calculated until the tubular network and cisternae have been extracted (See Chapter 9). Once this step is complete, results for

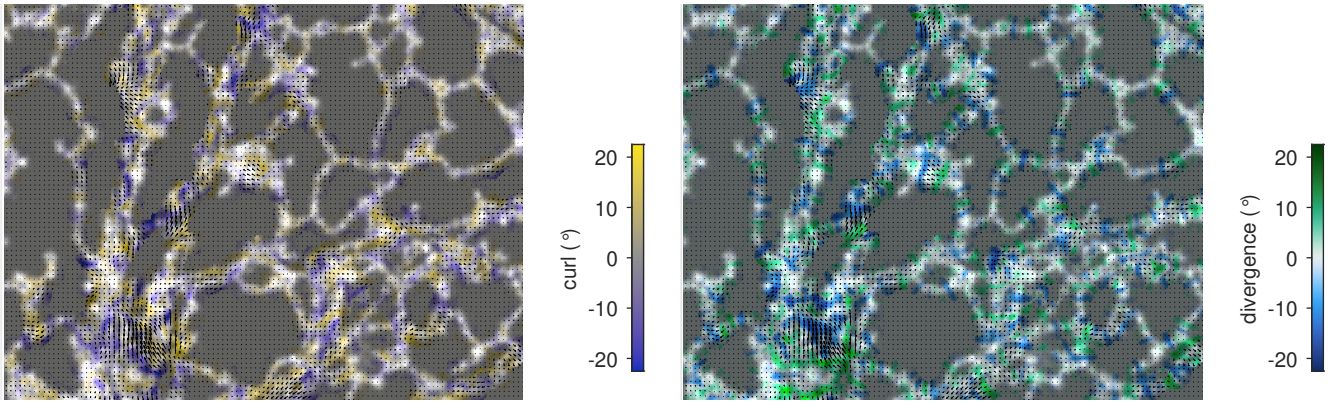
speed and direction are expressed as maximum, local (scalar) or global (vector sum) values for each tubule and cisterna, and can be displayed as a colour-coded overlay on the image (e.g. Fig. 7.7(b,c)).



(a) local velocity (b) maximum tubule speed (c) Direction of movement

Figure 7.7: Graphical overlay of average tubule speed and direction determined by the Farneback method

Other properties of the velocity vector field can be calculated to give an indication of the flow dynamics. Flow curl characterises the rotation normal to the flow velocity, whilst the flow divergence gives a measure of whether flow lines are diverging or converging (Fig. 7.8(a,b) and Fig. 7.9(a,b).

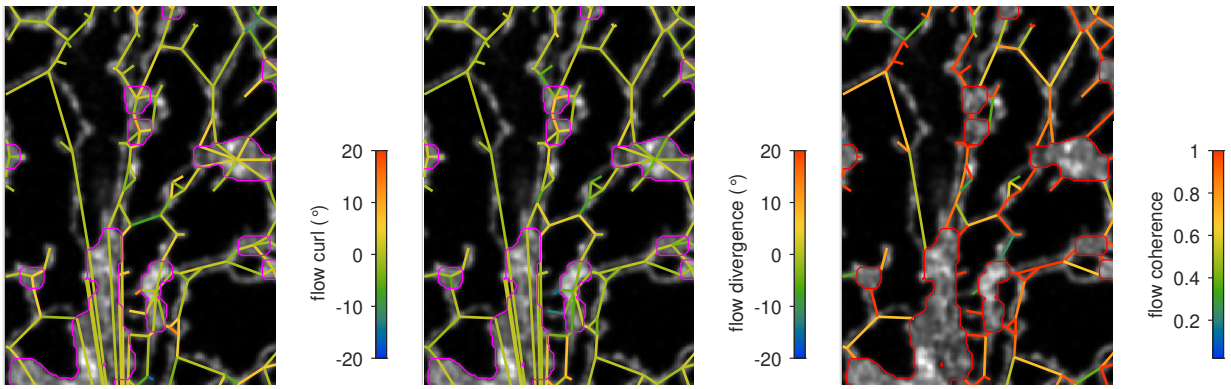


(a) Colour-coded optical flow (b) Quiver plot of velocity

Figure 7.8: Colour-coded representation of local curl and divergence

At the moment the significance of curl and divergence of the vector flow field is not clear as movement in the ER is a complex combination of diffusion, membrane flow and bulk flow, which do not directly correspond to conventional fluid flow dynamics. An additional measure, the flow coherence, is the ratio of the scalar sum of speeds at each pixel in the tubule to the vector sum of all pixels (Fig. 7.9(c)), may provide a more useful metric. A value of 1 indicates all the local flows are aligned in the same global direction. Values less than 1 indicate there is movement within the region, but it is not necessarily aligned. This may occur if there are shape changes rather than directed flows.

Results for any of the flow metrics can also be plotted as histograms (Fig. 7.10(a)) for individual frames or the whole image series using the plot controls (Fig. 7.10(b)). Distributions can be

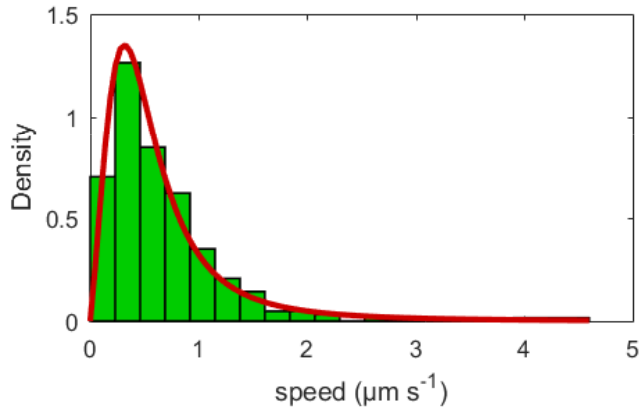


(a) Flow curl (b) Flow divergence (c) Flow coherence

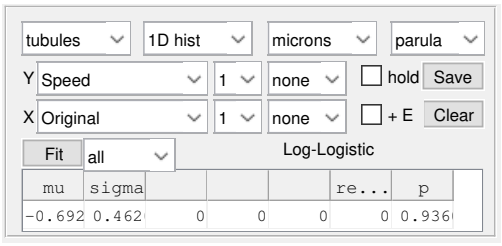
Figure 7.9: Graphical overlay of flow curl, divergence and coherence

Fit using a variety of probability distributions, selected from the adjacent dropdown menu. The *all* option automatically find the best fit to the data, reports the distribution selected and gives the distribution parameters in the table below.

Figure 7.10: Graphical representation of average tubule speeds



(a) distribution of tubule speeds



(b) plot controls

7.5 *Visualisation of the ER speed as a movie*

Changes in velocity (or indeed animation of any image type in a time-series) can be achieved within the network interface using the movie controls in the display panel (Fig. 7.11).

Figure 7.11: Controls to animate time-series images within the network analysis GUI

Alternatively the **Viewer** button opens a separate GUI that has more options to control display of the movie, including output to a variety of video formats (Fig. 7.12). The **Viewer** program is

described in more detail in Chapter 14.

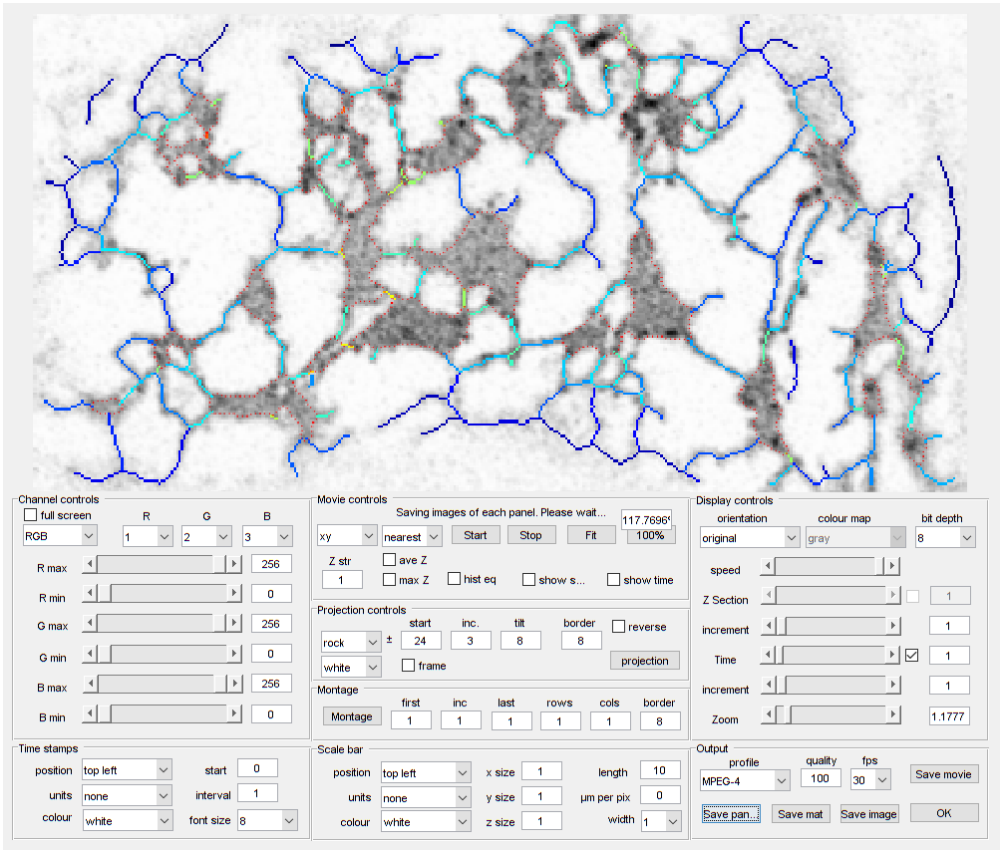


Figure 7.12: The movie viewer GUI

8

Persistency mapping

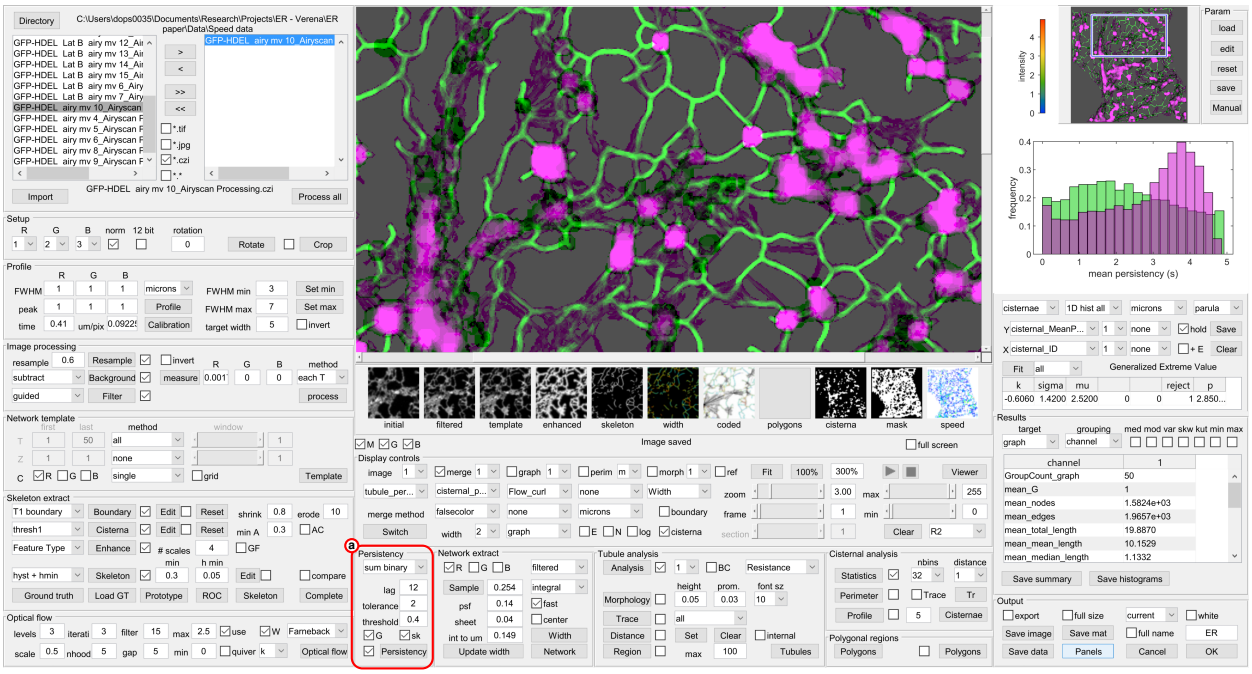


Figure 8.1: Persistency analysis controls

8.1 *Introduction*

Whilst elements of the plant ER network are highly dynamic, there are also components that remain static over a long period of time[1]. The static elements include regions where the ER is tethered to the plasma membrane at ER-plasma membrane contact sites (EPCS)[2], which act as fixed anchors to stabilise the polygonal network[3], as well as relatively immobile tubules and cisternae. Persistency mapping was developed to highlight these features, and has employed a variety of image processing steps to identify persistent, immobile structures.

At present there is no agreed definition of the duration an object has to remain in position to be regarded as a persistent feature, but current practice has adopted a period of 5-10s for tubules and cisternae, or longer for persistent nodes. Thus, in the initial pa-

[1] L. R. Griffing, C. Lin, C. Perico, R. R. White, and I. Sparkes. Plant ER geometry and dynamics: biophysical and cytoskeletal control during growth and biotic response. *Protoplasma*, 254: 43–56, 2017

[2] I. Sparkes, J. Runions, C. Hawes, and L. Griffing. Movement and remodeling of the endoplasmic reticulum in nondividing cells of tobacco leaves. *The Plant Cell*, 21:3937–3949, 2009b

[3] I. Sparkes, T. Ketelaar, N.C.A. De Ruijter, and C. Hawes. Grab a Golgi: laser trapping of Golgi bodies reveals in vivo interactions with the endoplasmic reticulum. *Traffic*, 10:567–571, 2009a

per (Sparkes *et al.* 2009) images separated by 8s intervals from a 50-frame time-course lasting 80s were subtracted, binarised, separated into tubules and cisternae by opening and closing operations, and then summed over the time course to give persistency maps. Summary statistics were calculated by normalising to the total membrane area present in the time-course. In subsequent analyses that focussed more on the biophysics of the polygonal network, persistent nodes were identified as local maxima after averaging over the complete time course and smoothing in (x,y) with a Gaussian filter, whilst the remainder of the ER network was thresholded and thinned to give a single pixel-wide skeleton[4], or segmented manually[5].

[4] C. P. Lin, Y. W. Zhang, I. Sparkes, and P. Ashwin. Structure and dynamics of er: minimal networks and biophysical constraints. *Biophysical Journal*, 107: 763–772, 2014

[5] C. Lin, R.R. White, I. Sparkes, and P. Ashwin. Modeling endoplasmic reticulum network maintenance in a plant cell. *Biophysical Journal*, 113:214 – 222, 2017

8.2 Implementation of persistency analysis

The **Persistency** control panel provides a number of different methods to find the persistent nodes and calculate persistency maps.

Maps of persistent tubules and cisternae can be calculated from either the background-subtracted intensity images or the segmented skeleton and cisternal images, selected using the drop-down menu.

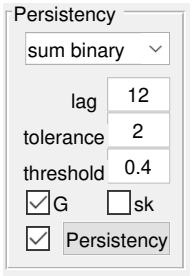


Figure 8.2: Controls to measure persistency of tubules and cisternae

- *sum binary* : dilates the pixel skeleton and the binary cisternal images by the **tolerance** (in pixels), and then calculates a moving sum over the **lag** period to give separate persistency maps for tubules and cisternae. The result is normalised by the lag period. A value of 1 means the pixel was occupied throughout the lag period.
- *diff binary* : dilates the pixel skeleton and the binary cisternal images by the **tolerance** and then determines whether a pixel was present at the start and end of the **lag** period using an AND operation. The result is normalised by the lag period. A value of 1 means the pixel was still occupied at the end of the lag period.
- *all binary* : follows the same procedure as *sum binary* except that the dilated skeleton and cisternal images are combined before processing to give a single persistency map. This ensures that persistent nodes that may remain over time as tubules and cisternae move are still identified.
- *sum intensity* : Calculates a moving sum over the **lag** period for the background-subtracted intensity time-series. An automated threshold is applied to segment the network from the time-averaged images. Persistent tubules are separated using closing and thinning, and persistent cisternae by opening. The binary skeleton and cisternal images from each time point are used to mask the results.
- *diff intensity* : Calculates a difference image for the first and last images of the **lag** period from the background-subtracted intensity time-series, and masks the result with the binary AND

image of the each pair of time-points. An automated threshold is applied to segment the network from the difference imaging, and then persistent tubules are separated using closing and thinning, and persistent cisternae by opening. The binary skeleton and cisternal images are used to mask the results.

The overall tubule persistency or cisternal persistency is calculated as the average over time, and can be visualised as a merged image between the map of persistent tubules (green) and persistent cisternae (magenta). If the **G** checkbox is ticked, the background set to grey to allow visual discrimination of regions of low persistency that are coded as black (Fig. 8.3). The **sk** checkbox superimposes the pixel skeleton for the first time-point in blue. Results for each lag period can be animated using the play button in the **Display**.

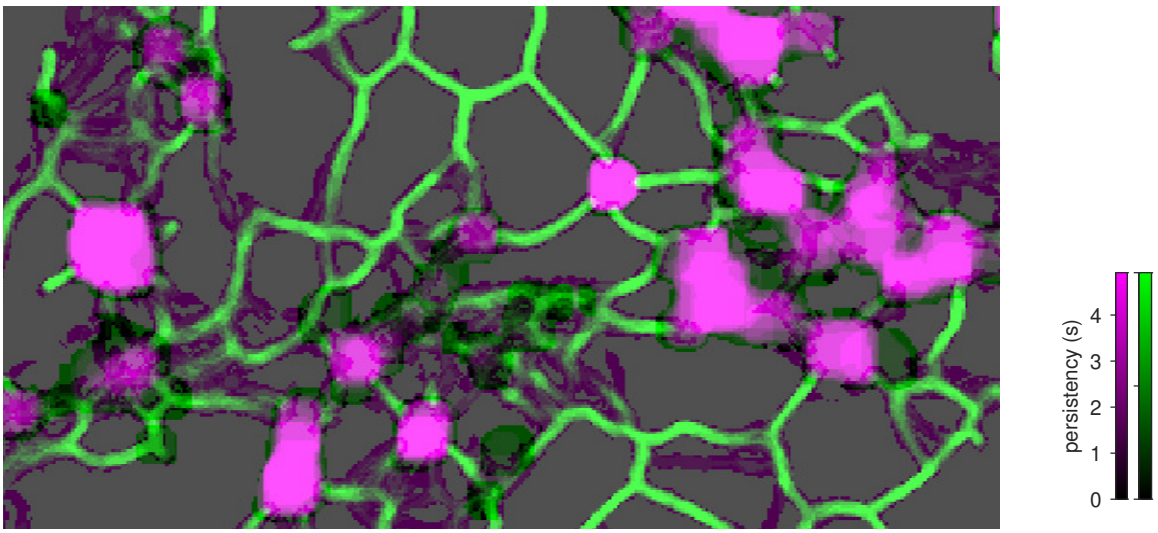


Figure 8.3: Persistency map of tubules (green) and cisternae (magenta)

8.3 *Identification of persistent nodes*

Persistent nodes may remain in place for extended periods even as tubules and cisternae flow over them. They are calculated in five steps from the background-subtracted time-series using a slightly modified version of Lin *et al* (2014)[6] :

- Filter in time using a 1-D median (rather than an average filter in time)
- Filter in space using Gaussian filter in (x,y) with sigma set to 0.1 μm
- Normalise the filtered image
- Set values below a **threshold** in the normalised image, typically around 0.3-0.5 (set by the **threshold** textbox), to zero. Persistent nodes tend to be brighter, so this provides some selection for more prominent nodes.

[6] C. P. Lin, Y. W. Zhang, I. Sparkes, and P. Ashwin. Structure and dynamics of er: minimal networks and biophysical constraints. *Biophysical Journal*, 107: 763–772, 2014

- Find local maxima in the masked image and overlay these on the image.

The persistent nodes are dilated by the tolerance to aid visualisaiton and overlaid on any one of the other ER persistency maps (magenta). Alternatively, the *persistency* image shows the persistent nodes overlaid on the green/magenta merge of tubule persistency and cisternal persistency, along with the original skeleton in blue (Fig. 8.4). This approach finds persistent nodes whether they are part of the tubules, cisternae or transition between the two over the time course.

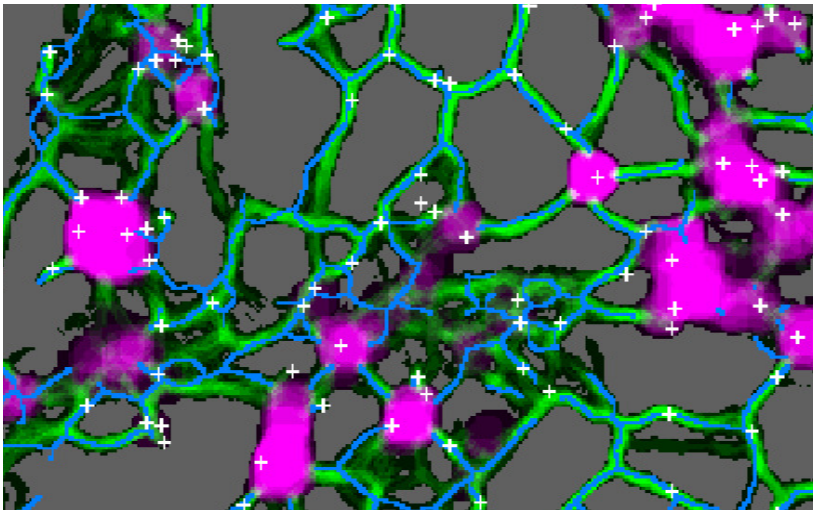
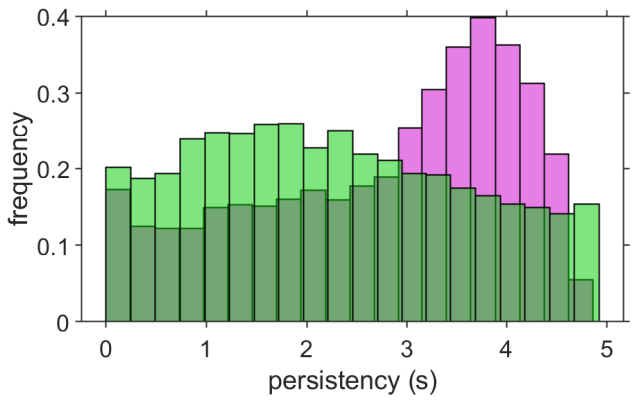


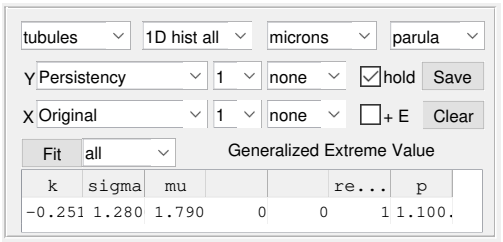
Figure 8.4: Combined persistency map of tubules and cisternae (green), overlaid with persistent nodes (magenta)

8.4 Graphical results for persistency

Histograms of tubule, cisternal or node persistency can also be plotted with the color corresponding to the merged persistency image (Fig. 8.5) for individual frames or the whole image series using the plot controls (Fig. 8.5(b)).



(a) persistency histograms



(b) histogram display controls

Figure 8.5: Histogram of tubule persistency (green) and cisternal persistency (magenta)

9
Extraction of a weighted network

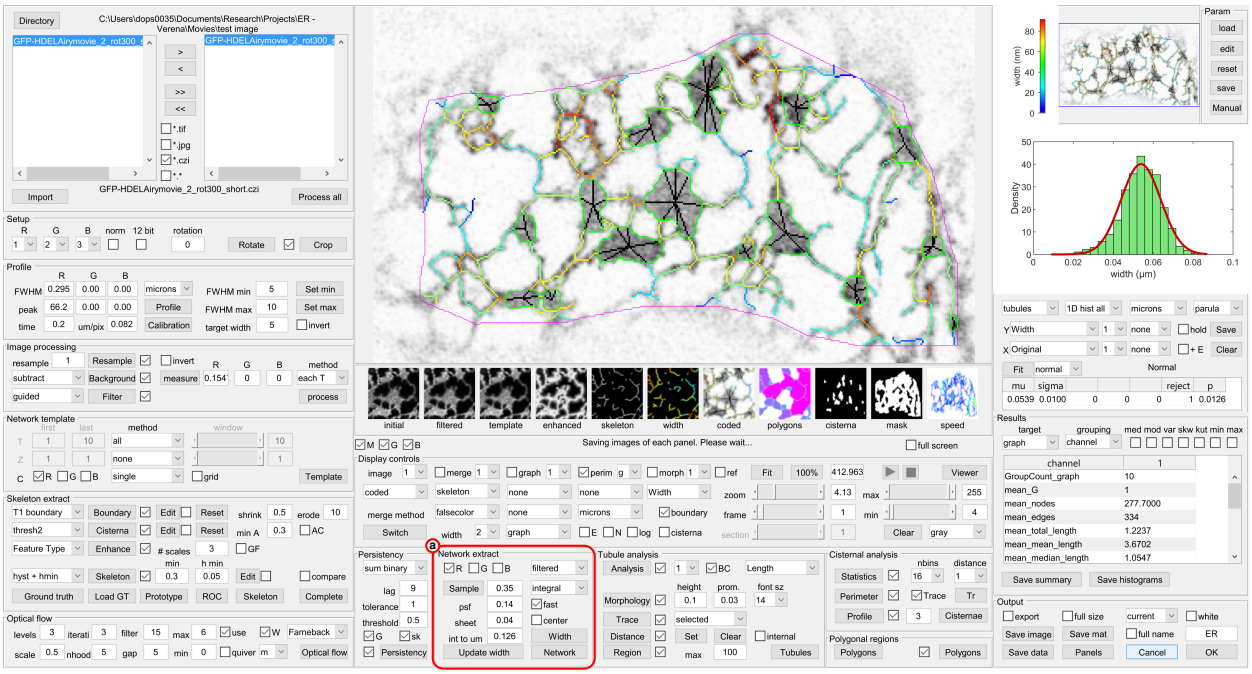
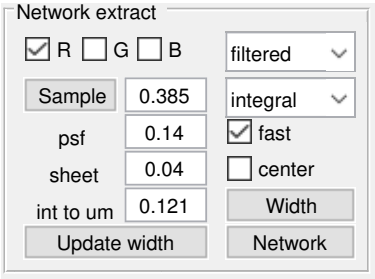


Figure 9.1: Controls to measure the tubule widths and extract a weighted network (a) and below



9.1 *Estimation of the tubule diameter*

Once the pixel-skeleton has been segmented satisfactorily, the next step in network extraction is to estimate the tubule width, before converting the pixel skeleton to a weighted graph representation. In a multi-channel image, one or more channels can be selected for analysis using the **R,G,B** checkboxes in the **Network Extract** panel (Fig. 9.1). The target image to base the width measurements can be selected from the adjacent dropdown menu, and would typically be the background-subtracted image, or the filtered image for additional noise-reduction.

Unfortunately, the expected ER tubule diameter (40 nm) is below the theoretical resolution of a confocal microscope, and at the resolution limit of current live-cell super-resolution techniques, such as STED[1], so direct physical estimates of the width are challenging.

[1] B. Hein, K. I. Willig, and S. W. Hell. Stimulated emission depletion (sted) nanoscopy of a fluorescent protein-labeled organelle inside a living cell. *PNAS*, 105:14271–14276, 2008

The initial confocal image and the methods described here typically return a value of the 'true' width convolved with the point spread function (psf) of the microscope. In addition, even sampling with a Nyquist pixel spacing based on the psf, can lead to significant digitisation errors during processing, as the size of the tubules is only a few pixels wide. Up-sampling the image (as part of the resampling process) can help to reduce these digitisation errors.

A number of different approaches are available that all provide some information on the tubule diameter including:

- *'Distance'* : This estimates the local FWHM of the tubule at each pixel in the skeleton using the original tubule intensity. The peak intensity is estimated from the background-subtracted image, sampled for each pixel in the skeleton, whilst the distance is estimated from the distance transform of the pixel skeleton. The 50% threshold is estimated from where the pixel intensity falls below half the peak intensity, assuming a local background of zero. At the moment there is no interpolation, so there is likely to be considerable discretisation error with this approach, although this will be partially compensated by the averaging that takes place later along each tubule during graph conversion to give the mean width of each tubule.
- *'Maximum-gradient granulometry'* : The intensity image is subject to a series of image openings (erosion followed by dilation) that progressively remove structures as the size of the opening kernel exceeds the size of the underlying object. This results in an intermediate (*x*,*y*,*s*) image, where *s* increases with the size of the disk-shaped kernel. The intensity of each pixel in the skeleton initially decreases slowly with *s* as the kernel samples more of the object, but then reduces dramatically once the boundary of the object is reached, and the kernel only samples the background. The transition point for any pixel is determined from the maximum (negative) gradient of the granulometry curve. This approach constrains the width to integer pixels values, and also suffers from the digital approximation of small kernels to a true disk shaped kernel.
- *'Integrated intensity granulometry'* : This approach follows the same methodology as the maximum-gradient granulometry method, but rather than extract a specific size threshold, the integrated intensity under the granulometry curve is calculated. This provides a more nuanced interrogation of the local image intensity, but cannot be directly related to the physical tubule width without additional assumptions about the relationship between fluorescence intensity and sampled volume. This approach does help with estimation of relative tubule widths, even if they are sub-resolution objects, provided it is assumed that the fluorescent probe is evenly distributed throughout the ER, and the ER is within the sampling volume of the confocal defined by the psf.

This principle behind the integrated intensity measurements can be illustrated empirically using a simulated model of ER tubules with different widths (Fig. 9.2A,B) that are convolved with the excitation and emission psfs, modelled as anisotropic 3-D Gaussian[2] blurring functions (Fig. 9.2C,D).

[2] B. Zhang, J. Zerubia, and J.-C. (2007). Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Applied Optics*, 46:1819–1829, 2007

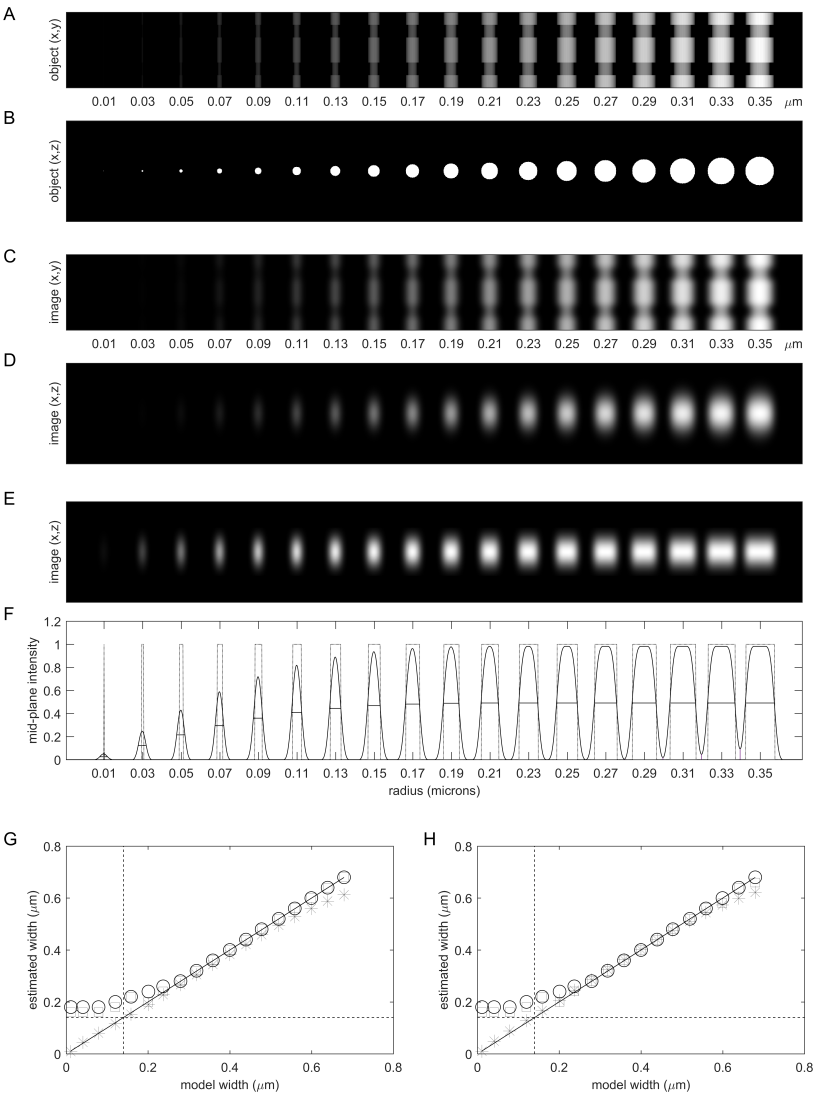


Figure 9.2: Estimation of sub-resolution tubule widths. A set of ER tubules with increasing width were simulated as cylinders and viewed as an average (x,y) projection (A,C) and a single (x,z) section (B,D) before (A,B) and after (C,D) blurring with a simulated anisotropic 3-D Gaussian excitation and emission point spread function (FWHM $0.14\mu m \times 0.14\mu m \times 0.42\mu m$ in x,y and z, respectively). The simulated result for scanning a single confocal section is illustrated in (E). The intensity profile across the mid-section of the original tubules and after blurring is shown in (F). The reduction in intensity for sub-resolution tubules is apparent. (G) shows the performance of the three different estimates of tubule width. The 50% distance measure (o) and the maximum gradient granulometry (square) perform well above about twice the psf, but return a value close to the psf for sub-resolution objects. The integrated intensity measure (x) gives a near linear response throughout the range of tubule widths examined. Similar result are obtained using a faster granulometry method that includes decomposition of the circular structuring element into linear approximations, along with an empirical correction of -1 pixel to compensate for oversampling (H).

The performance of the three different width estimators can be assessed by analysis of a single confocal plane to represent an (x,y) image plane (Fig. 9.2E). All measures perform well above about twice the theoretical lateral FWHM of the psf (0.14μm for the Zeiss AiryScan), but the 50% distance measure and the maximum-gradient granulometry measure both converge to a fixed value close to the psf for sub-resolution objects as expected (Fig. 9.2G). By contrast, the integrated intensity measure provides near linear performance even with sub-resolution objects (Fig. 9.2G). An additional option is available to increase the speed of the granulometry by using decomposition of the disk-shaped structuring element into a set of linear elements. This requires an empirically determined

correction of -1 pixel to the estimated width to compensate for the over-sampling with essentially square kernel approximations (Fig. 9.2H).

Whilst this suggests that the integrated intensity measurement is linear with the size of the objects, it still requires knowledge of a calibration factor to relate intensity to radius. To compare with the other approaches to width measurement that provide an output in pixels, we need to estimate the fluorescence from a known volume of ER.

Perhaps the simplest measure is the average intensity for a cisternal sheet, which represents the signal expected for a structure completely filling the (*x*,*y*) plane of the estimated psf, but with a similar thickness to the tubules. This can be samped from the background corrected and filtered image using the **Sample** button (Fig. 9.3),which also updates the value in the adjacent textbox.

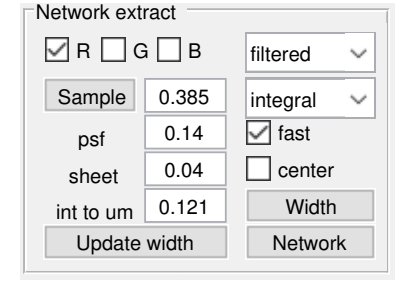


Figure 9.3: Controls for width calbration

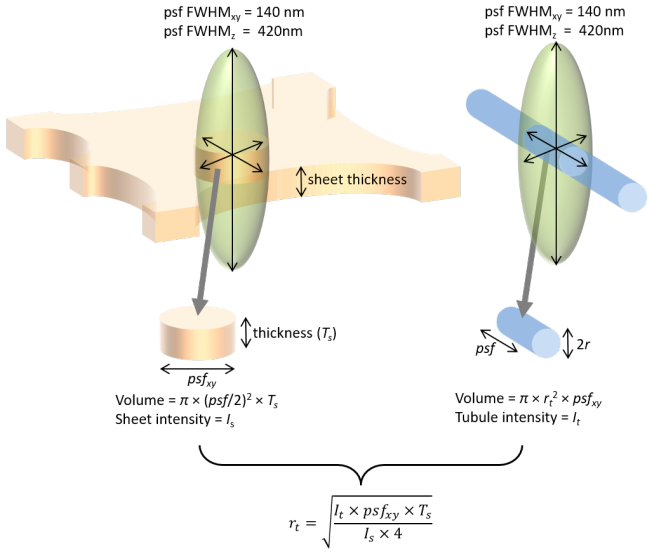


Figure 9.4: Approximate volumes, and hence relative fluorescence intensity, sampled by the psf across a cisterna and a tubule. The correction factor is based on measurement of the sheet intensity (I_s) an estimate of the sheet thickness (T_s), and an estimate of the lateral resolution of the point spread funtion (psf_{xy})

The corresponding volume sampled is calculated using an estimate of the $FWHM_{xy}$ of the psf, entered in the **psf** textbox, and an estimate of the sheet thickness, entered in the **sheet** textbox. For a typical image collected without saturation, the average sheet intensity is around 0.3-0.5, whilst $FWHM_{xy}$ of the AiryScan psf is taken as 0.14 μm, and the sheet thickness (T_s) as 0.04 μm. This gives a typical correction from intensity to microns of 0.1-0.15 $\times \sqrt{I_t}$, where I_t is the integrated intensity of the tube. (Fig. 9.4).

The width estimate for each tubule is calculated by clicking the **width** button and is presented as a pseudo-colour coded version of the pixel skeleton (Fig. 9.1) that is scaled from the minimum width in blue to the maximum in yellow (depending on the look-up table chosen in the display panel). The colour-bar is updated to display the estimated width in microns.

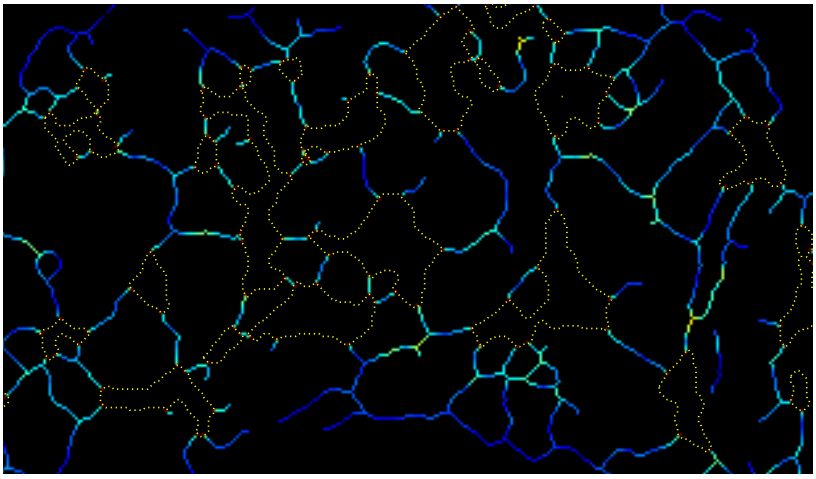


Figure 9.5: Pseudo-colour coded representation of the tubule width (in microns). The cisternal regions are shown as a dotted yellow outline

9.2 *Conversion of the pixel skeleton to a graph representation*

The **Network** button will trigger all the subsequent steps required to convert the pixel-skeleton to a weighted graph representation. For the tubular regions of the skeleton, the nodes are defined at the junctions between the ER tubules, or the end-points of free tubules, which are connected by straight edges that preserve the topology of the network (Fig. 9.7). Any loops connecting the same two nodes, will be split to create an extra node with degree 2 in the longer arm (Fig. 9.6). This ensures that metrics that are critically dependent on parameters such as resistance to flow are correctly assigned to each edge separately, rather than to an average of the two edges. The cisternae are represented as a super-node placed at the centroid and connected to all the tubules incident on the boundary.

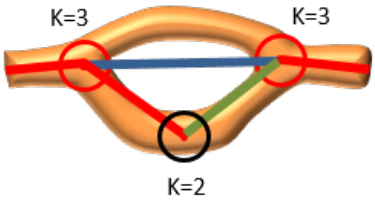


Figure 9.6: Resolution of loops into two arms by inserting an additional k=2 node

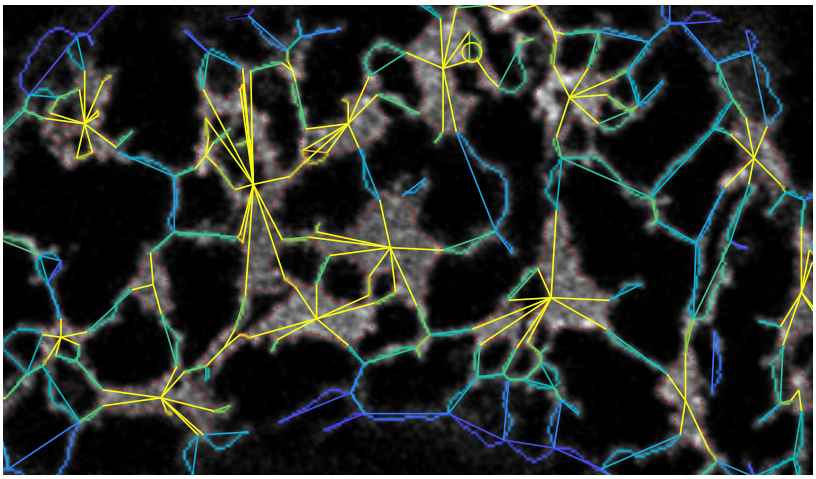


Figure 9.7: Conversion of the weighted pixel skeleton to a weighted graph. Junctions and free ends are represented as nodes linked by edges that have a vector of properties associated with them including width and length. Cisternae are represented as a 'super-node' connected to all the incident tubules on the boundary

A number of metrics are associated with each edge including the Euclidean length of the underlying pixel skeleton, the average

width, and the average width of the tubule excluding the nodes (termed *'center width'*), which provides a more accurate measure of the tubule diameter itself (Fig. 9.8). The precise number of pixels to ignore in estimation of the width is currently set to be the average width of the edges incident at the node, unless the node has a degree of two or less and is not connected to a cisternal sheet.

If the **graph** checkbox is ticked in the display control panel (Fig. 9.9(a)), the metric to be displayed is selected from a drop-down menu (Fig. 9.9(b)), and overlaid on the image as a graph, with the nodes at the junctions connected by straight edges that match the colour-coding of the skeleton (Fig. 9.9(d)) or a pixel skeleton (also shown in Fig. 9.7). Various metrics for the nodes can also be displayed as colour-coded points using the (Fig. 9.9(c)) drop-down menu. The actual values for the edges and nodes can be shown using the **E** and **N** checkboxes, respectively, the scaling can be changed to logarithmic with the **log** checkbox (Fig. 9.9(e)).

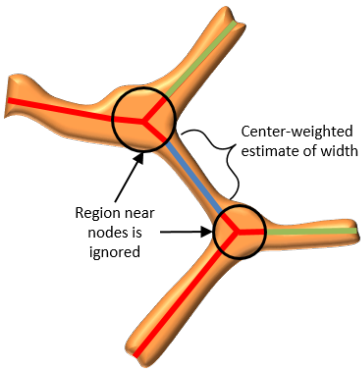


Figure 9.8: Center-weighting of the tubule width ignores the contribution of pixels near the nodes

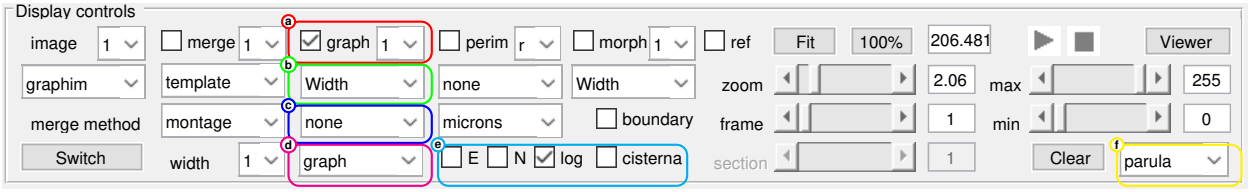


Figure 9.9: Controls that affect display of the weighed graph

The colorbar is updated to reflect the calibrated metric (Fig. 9.10), and the color-map can be changed using the **colormap** drop-down menu (Fig. 9.9(f))

If any ER cisternae have been identified, they are represented as a 'super-node' positioned at the intensity-weighted centroid position that is connected to all the tubules that are incident on the cisternal boundary. Each of these tubes is given an arbitrary value equal to the average width value. The boundary of the cisternae can be highlighted using the **perim** checkbox (Fig. 9.9(e)).

The graph overlay can be removed by clicking the **Clear** button.

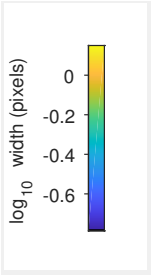


Figure 9.10: The calibrated width colorbar

9.3 *Graphical output*

Metrics for the tubules and nodes can be plotted in pairwise combinations (Fig. 9.3(a)), individually as a histogram (Fig. 9.3(c)), or as an average value over time (Fig. 9.3(d)), depending on the settings in the plot controls panel (Fig. 9.11).

The first dropdown menu (Fig. 9.11(a)) selects the object to plot (*tubules, nodes, etc.*), whilst the second (Fig. 9.11(b)) selects between a *scatter, 1D histogram, 2D histogram* or *time* plot in pixels or microns selected by Fig. 9.11(c). The metric to be plotted on each axis is selected from the Y and X dropdown menus (Fig. 9.11(d,e)), for a specific channel (default as channel 1). The data can be transformed using the adjacent dropdown menu (*log, log10, inverse, sqrt, exp, arcsin, logit*). In the case of log transforms, the **+E** checkbox adds

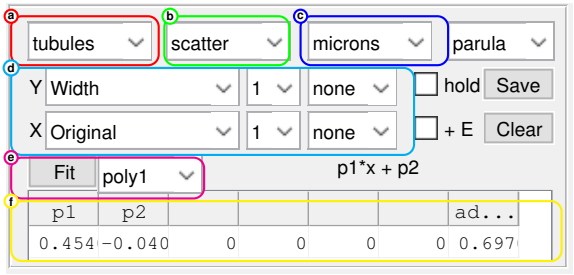
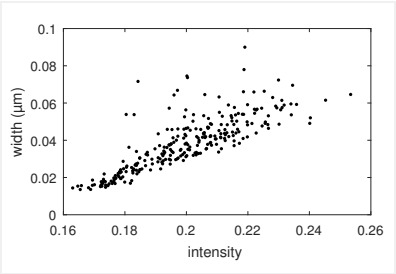
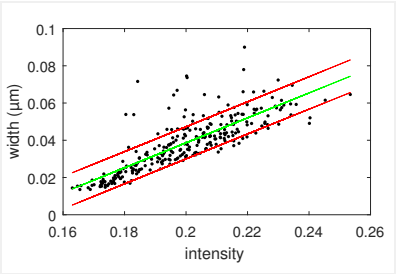


Figure 9.11: Plot controls for graphical output for the network metrics

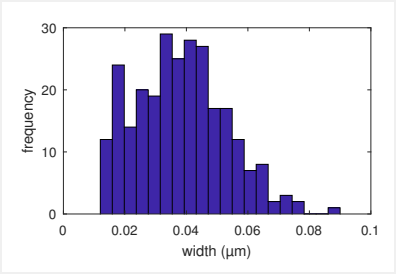
half the minimum value to any zero entries if required to avoid log(0) errors. In addition, the scatter plot data can by fit with a variety of functions (*polynomial* up to order 3, single or double *exponential*, and single or double *power*) using the **Fit** button (Fig. 9.3(c)), with the coefficients returned in the table, along with an adjusted r^2. Multiple plots can be overlaid using the **hold** checkbox. In the case of histograms for time-series, the colour is controlled by the look-up table chosen. Plots can be saved as images or cleared using the **Save plot** and **Clear** buttons, respectively.



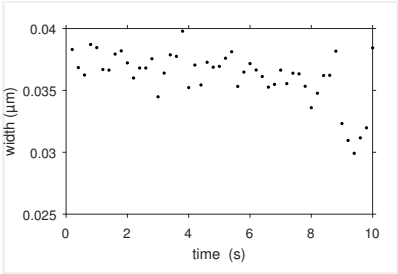
(a) Scatter plot of tubule width against intensity



(b) Data fit with a linear regression



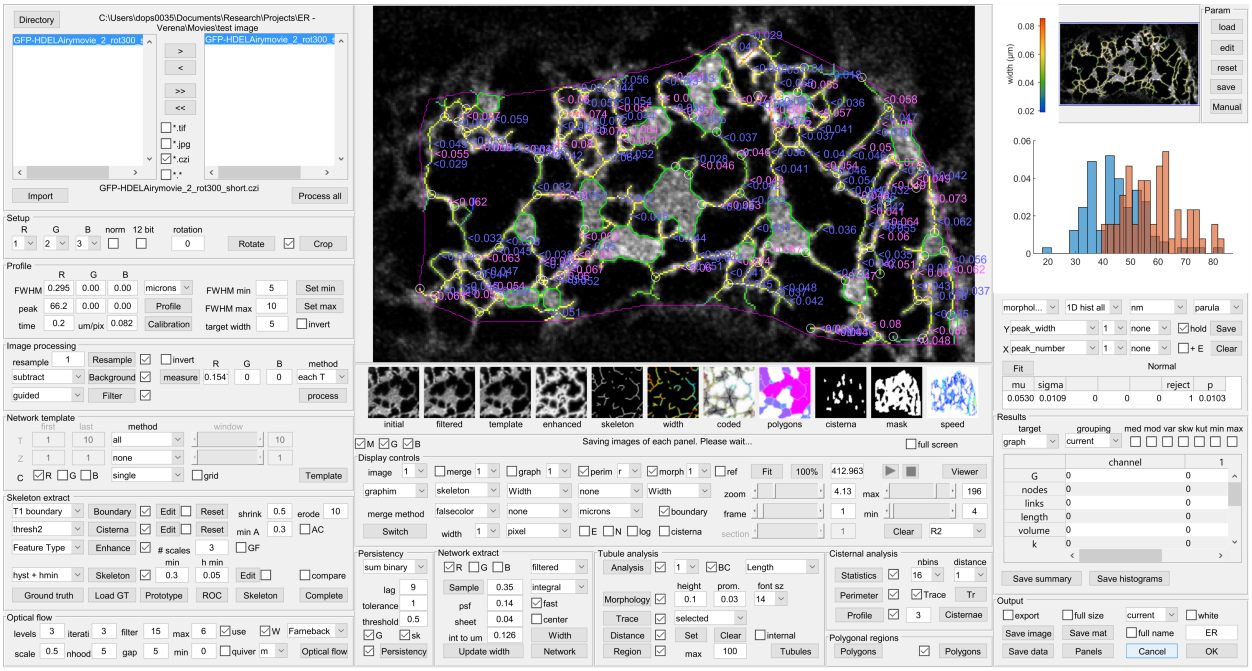
(c) Histogram of tubule widths



(d) Average width plotted against time

10

Analysis of the ER tubule structure



The **Analysis** button in the **Tubule analysis** panel (Fig. 10.1(a)) calculates a set of quantitative metrics for each tubule by using the pixel co-ordinates in each edge to extract corresponding values from the width, speed and persistency images (Table 10.1).

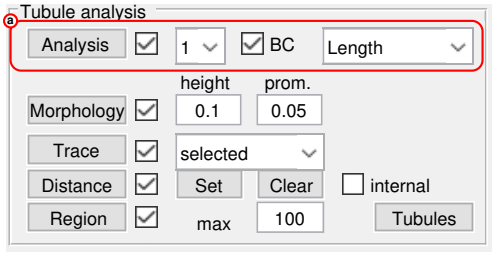


Figure 10.1: Controls for graph analysis of the tubular network

In addition, a number of graph-theoretic metrics are calculated for the network of tubules (Table 10.2), excluding the cisternal nodes and cisternal perimeter, for each time-point, that are displayed in the **Results** panel (Fig. 10.2). These metrics include the

Table 10.1: Morphological metrics calculated for each ER tubule

Metric	**Explanatory notes**
Original, cv	The mean intensity of pixels in the edge from the original image, along with the coefficient of variation (cv)
Intensity, cv	The mean intensity and cv after background subtraction
Width, Width cv	The estimated width ($2r$) and cv (μm)
Width center	The estimated width excluding overlap regions at the node
Length, length center	The total length and length excluding the overlap at nodes (l, μm)
Area, Volume	The cross-sectional area ($a = \pi r^2$, μm^2) and volume ($v = a * l$, μm^3)
Resistance	The predicted resistance to flow assuming Poiseuille flow (l/r^4, μm^{-3})
Number	The number of pixels in the edge
Tortuosity	The Euclidean distance between the nodes divided by the total length of the edge
Speed - local	The scalar sum of the speeds calculated for each pixel ($\mu m\ s^{-1}$)
Speed - max	The maximum speed for any pixel ($\mu m\ s^{-1}$)
Speed - global	The vector sum of speeds for each pixel ($\mu m\ s^{-1}$)
Dir - local, std	The angular mean and standard deviation of the flow direction at each pixel ($^\circ$)
Dir - global	The direction of the vector sum ($^\circ$)
Flow coherence	The ratio of the local speed (scalar sum) to the global speed (vector sum)
Flow curl, std	The local angular rotation of the vector field ($^\circ$)
Flow AV, std	The local angular velocity of the vector field ($^\circ s^{-1}$)
Flow divergence, std	The divergence of the vector flow field ($^\circ$)
Persistency, cv	The mean period of time that each pixel forms part of a tubule (s)
Distance	The geodesic distance of each pixel from a manually defined reference point (μm)
Region	The integrated sum of intensities in the region closest to each edge
BC	The edge betweenness centrality

number of nodes and edges and their length and width. The Global and Root efficiency is calculated[1], as the sum of the inverse of shortest paths between all nodes, or the first cisternal node, respectively.

[1] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Phys. Rev. Lett.*, 87:198701, 2001

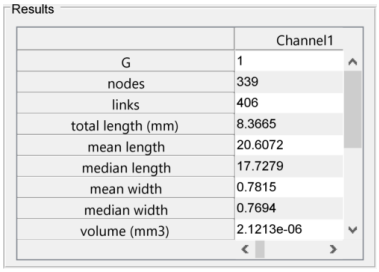


Figure 10.2: Graph-theoretic measures of the tubular networkk

Metric	Explanatory notes
G	The number of sub-graphs in the network. This is typically one as only the largest connected component is selected
nodes	The number of nodes (excluding cisternal nodes)
links	The number of edges (excluding cisternal edges)
total length	The total length of the network tubules (mm)
mean length	The average length of the tubules (μm)
median length	The median length of the tubules (μm)
mean width	The average width of the tubules (μm)
median width	The median width of the tubules (μm)
volume	The total volume of the network (mm^3)
k	The average node degree
G efficiency	The global efficiency of the network
R efficiency	The root efficiency of the network calculated to the first cisternal node
cyclomatic no.	The cyclomatic number
alpha	The alpha coefficient or meshedness
beta	The beta coefficient
gamma	The gamma coefficient
diameter	The longest shortest path through the network

Table 10.2: Morphological metrics calculated for each ER tubule

The global efficiency[2] is defined as the mean reciprocal of the shortest paths, weighted by resistance, with the reciprocal for disconnected nodes defined as zero (Equation 10.1). The root efficiency (E_{root}) is calculated in a similar manner but just from the exit point to all other nodes.

[2] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Phys. Rev. Lett.*, 87:198701, 2001

$$E_{global} = \frac{1}{N(N-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}} \tag{10.1}$$

The cyclomatic number, alpha, beta and gamma metrics derive from transportation geography for planar networks. The cyclomatic number gives a measure of the complexity of the network and is calculated as:

$$\text{cyclomatic number} = u = e - v + p \tag{10.2}$$

Where e is the number of edges, v is the number of vertices or nodes, and p is the number of components (typically set to 1 here as only the largest connected component is considered). The alpha coefficient[3] or meshedness[4] estimates the fraction of cycles present compared to a fully connected planar network, and is calculated as:

[3] P Haggett and RJ Chorley. *Network Analysis in Geography (pp 74-76) Edward Arnold Publishers Ltd.* London, 1969

[4] J. Buhl, J. Gautrais, R.V Solé, P. Kuntz, J.-L. Valverde, S.and Deneubourg, and G. Theraulaz. Efficiency and robustness in ant networks of galleries. *Eu. Phys. J. B*, 42:123–129, 2004

$$\text{alpha coefficient} = \alpha = \frac{e - v + p}{2v - 5} \tag{10.3}$$

The beta coefficient measures the level of connectivity in a graph based solely on the number of edges and vertices. Simple branching networks with no loops have β value of less than one, for a single loop $\beta = 1$, whilst $\beta > 1$ for more complex networks with multiple loops.

$$\text{beta coefficient} = \beta = \frac{e}{v} \tag{10.4}$$

The gamma coefficient represents the number of links present compared to the maximum number expected for a planar graph with the same number of vertices. Gamma ranges from 0 to 1 for a fully connected network.

$$\text{gamma coefficient} = \gamma = \frac{e}{3(v-2)} \tag{10.5}$$

In addition, if the **BC** checkbox is ticked, the *Betweenness Centrality* of the edges and nodes is calculated, using the metric in the adjacent dropdown menu, drawn from *Length, Width, Area, Volume* or *Resistance* to weight the shortest path calculation.

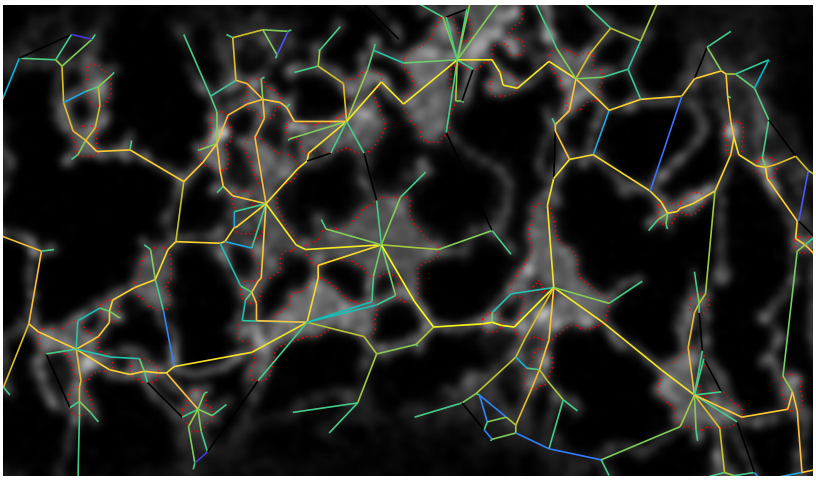


Figure 10.3: Notional importance of edges in the network, judged from their log Betweenness Centrality. Connections to the cisternal nodes are included in the calculation to ensure the network is fully connected

Betweenness centrality may provide some indication of likely routes of movement within the central part of the network. However, as the network is incomplete because typically it only contains a 2-D subset of a 3-D network, shortest-path measures on the network may be difficult to relate to a specific biological function.

10.1 Node metrics

A variety of metrics are also calculated for each node (Table 10.3)

The node degree is one for tubules with a free end, but typically 3 for all other nodes, excluding the cisternal nodes. Classifying the three main tubules as major (*Maj*), middle (*Mid*) and minor (*Min*) allows calculation of additional metrics such as the ratio of incident tubule widths, the mean orientation, and particularly the branch angles. For the latter two measures, the tubule is represented as

Table 10.3: Metrics calculated for each node

Metric	Explanatory notes
node degree	The number of tubules connected to each node, excluding cisternal nodes
total width	The sum of the tubule widths incident at each node
mean width	The average of the tubule widths incident at each node
intensity	The original intensity at the node
persistency	Node persistency
speed	Node speed
direction	Node direction
distance	The distance from the reference point
Maj, Mid and *Min*	The widths of the three main tubules incident at the node. Only one value is given for a terminal node, whilst few node have a degree greater than three
Min_Maj, Mid_Maj and *Min_Mid*	The ratio of the tubule widths for the three main incident tubules
Omaj, Omid and *Omin*	The orientation of the three main incident tubules, determined as a linear segment between the node and the midpoint of the tubule
Omin_Omaj, Omid_Omaj and *Omin_Omid*	Branch angles between the three main incident tubules, determined from a linear segment from the node to the midpoint of each tubule

a linear vector from the node to the mid-point of the tubule. This ensures that a reasonable length is considered when estimating the direction, but will incur errors for long tubules with high tortuosity.

10.2 *Tubule morphology*

Under some conditions or in certain genetic backgrounds, such as over-expression of reticulons[5], the ER tubules show fluctuations in diameter along their length. The **Morphology** button can be used to characterise the number, size and distribution of these bulges and constrictions (Fig. 10.4(b)).

[5] E. Breeze, N. Dzimitrowicz, V. Kriechbaumer, R. Brooks, S.W. Botchway, J.P. Brady, C. Hawes, A.M. Dixon, J.R. Schnell, M.D. Fricker, and L. Frigerio. A c-terminal amphipathic helix is necessary for the in vivo tubule-shaping function of a plant reticulon. *PNAS*, 113:10902–10907, 2016

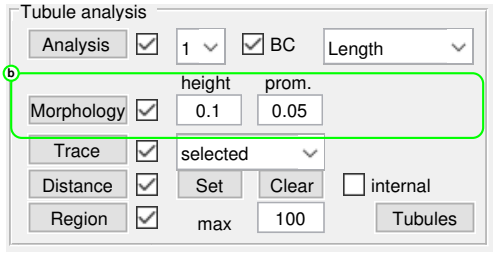
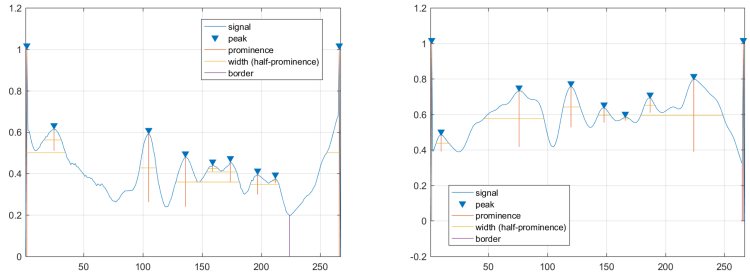


Figure 10.4: (b) Controls to analyse morphological bulges and constrictions along individual tubules

The pixel intensities are scanned along the length of each tubule starting at the nodes (which are set to 1), and the position and intensity recorded for each peak (bulge) that is greater than the minimum peak height set by the **height** textbox, and the minimum peak prominence set by the **prom.** textbox (Fig. 10.5). The prominence is the local height of a peak compared to its neighbours, rather than the baseline given by the peak height alone.

An additional minimum separation condition is imposed to prevent adjacent noisy pixels from being considered as separate peaks, and to exclude the original nodes from the analysis. This



(a) detection of intensity peaks (bulges) along an ER tubule

(b) detection of intensity troughs (constrictions) along an inverted ER profile

Figure 10.5: Tubule morphology: The intensity profile along each ER tubule is scanned to detect peaks (bulges) above a minimum intensity that are also a minimum height above their neighbours. In addition peaks have to be greater than a minimum separation from each other and the nodes (set to one). (b) The position of the 'constrictions' between the peaks are determined from a scan of the inverted intensity profile

separation defaults to $FWHM_{min}$. A similar scan is used on the inverted tubule profile to detect the *'troughs'* or constrictions. Once the location of the peaks and troughs has been determined, the width at each peak and trough is extracted, along with the *separation* between the peaks, if there are more that 2 peaks or troughs per tubule.

The position and intensity value of the peaks and troughs are overlaid on the currently displayed image with right justified labels, peaks in magenta and troughs in blue for channel 1 (Fig. 10.6).

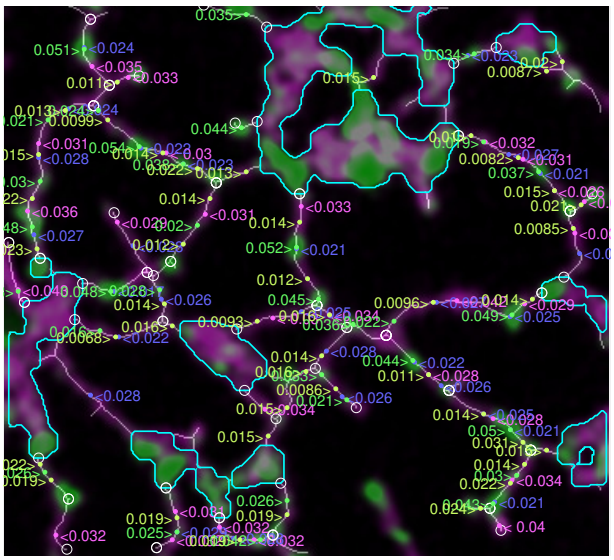


Figure 10.6: Measurement of peaks and troughs in the ER network. The estimated width for every pixel in each edge is scanned to detect local peaks, corresponding to bulges (magenta) and troughs, corresponding to constrictions (blue) for a lumenal marker. These are annotated with the estimate tubule width at that point for channel 1. Peaks and troughs are also detected for channel 2 representing an RFP labelled reticulon, and coded as green and yellow, respectively

If a second channel has also been analysed, peaks and troughs for this one are plotted with left justified labels, bulges in green and troughs in yellow.

The metric plotted can be selected using the **morph** dropdown menu (Fig. 10.7), from a selection that includes *Width, Covariance, Ratio* and *Value*, for the channel selected. The *Covariance* and *Ratio* values are only relevant if two channels have been analysed.

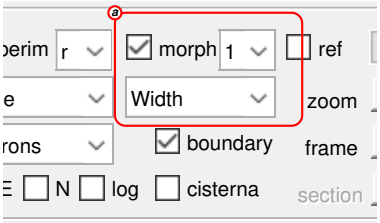


Figure 10.7: Controls to display the results of the tubule morphology analysis

10.3 *Tubule morphology traces*

The **Trace** button provides an alternative visual representation of the tubule morphology if the adjacent dropdown menu is set to *all*.

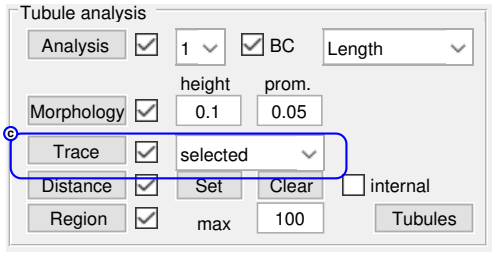
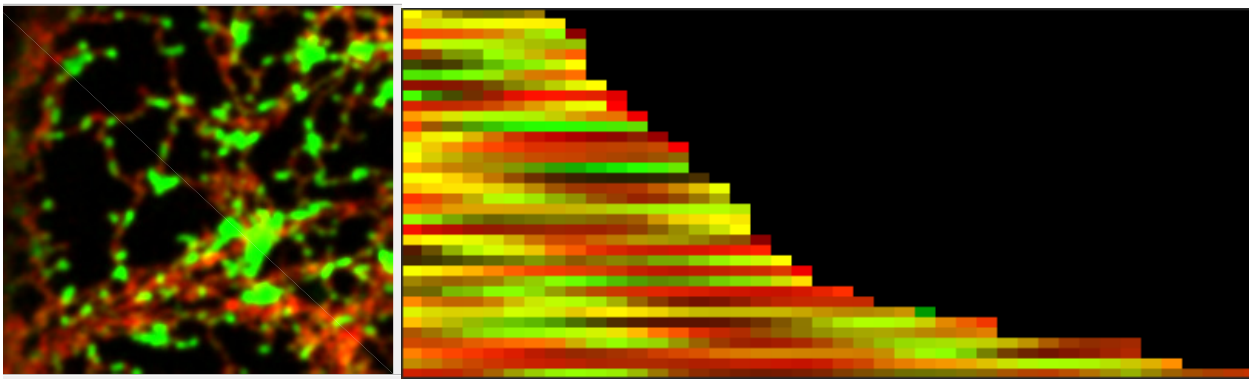


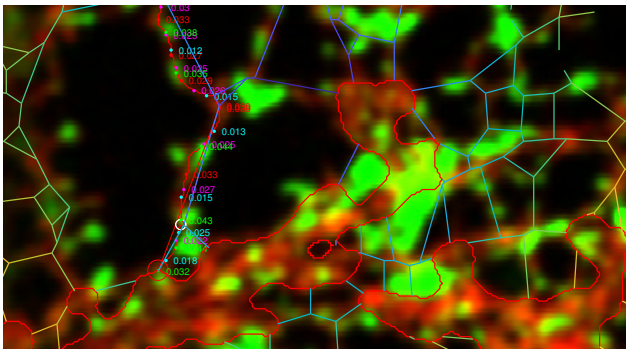
Figure 10.8: (c) Trace controls to extract profiles from all tubules or a selected tubule

This extracts the intensity profile for each channel for each tubule and orders them as successive left-justified lines. The resultant image is shown in separate image window (Fig. 10.9).

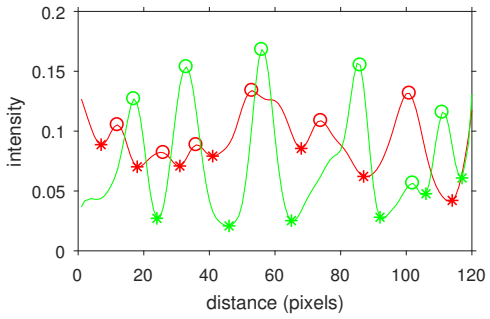


(a) Dual-channel image

(b) Stacked profile image



(c) Selected trace



(d) Selected trace plot

Figure 10.9: (a) Dual-channel image of GFP-HDEL (green) and reticulon (red). (b) aligned intensity profiles for each tubule, ordered by tubule length. (C) selected tubule, with the peaks and troughs labelled. (D) profiles along the selected tubule

If the tubule thickness is the same in each channel, the overall colour of the line will be yellow, otherwise peaks and troughs are visible as bands of green and red colour.

Results for a specific tubule can be obtained by setting the dropdown menu to *selected*, and then using the cursor to pick the start and end nodes from the graph displayed. The profile along the shortest path between the selected nodes is extracted and analysed for the peaks and troughs.

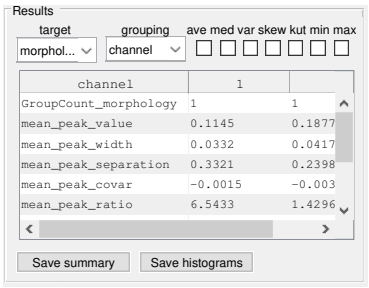
10.4 Summary of tubule results

The full set of metrics for the tubule morphology are given in Table 10.4.

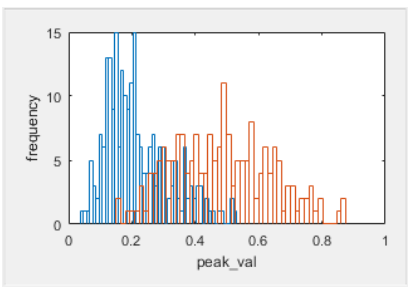
Table 10.4: Characterisation of tubule morphology

Metric	Explanatory notes
peak number	The total number of peaks measured excluding nodes. Values are given for each channel analysed
peak density	The number of peaks per unit length (μm^{-1})
length	The total length of tubules analysed (μm)
peak value	The mean intensity at the peak (I_P)
peak width	The estimated width of the peak (μm)
peak separation	The mean separation of peaks, which is only calculated if there is more than one peak per tubule (μm)
peak covar	The mean covariance of intensities from two channels measured at the peak ($(I_{P1} - \overline{I_{P1}}) * (I_{P2} - \overline{I_{P2}})$). Positive values indicate both channels show similar behaviour, whilst negative values indicate they move in opposite directions
peak ratio	The mean ratio of intensities at the peak from two channels (I_{P1}/I_{P2})
trough number	The number of troughs (measured as peaks in the inverted intensity profile)
trough density	The number of troughs per unit length of tubule (μm)
trough value	The mean intensity in the trough (I_T)
trough width	The mean width of the trough(μm)
trough separation	The mean separation of troughs if there is more than one trough per tubule (μm)
trough covar	The mean covariance of intensities from two channels measured in the trough ($(I_{T1} - \overline{I_{T1}}) * (I_{T2} - \overline{I_{T2}})$)
trough ratio	The mean ratio of intensities from two channels measured in the trough (I_{T1}/I_{T2})

The summary results for all the tubules are shown in the **Results Summary** panel (Fig. 10.10 (a)), or can be plotted as histograms (Fig. 10.10 (b)).



(a) Summary results for the tubule morphology measurements



(b) Histogram plots for the peak and trough intensities

Figure 10.10: Tubule morphology results : (a) Average values for tubule morphology metrics are shown for each channel. (b) histogram display of the distribution of bulge and constriction intensities

10.5 Distance measurements from a reference point

In some instance it is useful to map the morphology and dynamic of the ER against a fixed spatial reference point, such as the site of a

fungal hyphal penetration peg, a local region of photobleaching, or a localised wounding response.

The **Set** button (Fig. 10.10 (c)) allows the user to manually define a reference point on the image, which is saved with the experiment. The manual reference point can be removed using the **Clear** button, in which case it defaults to the center of the image. If the **internal** checkbox is ticked, clicking the **Distance** button will calculate the distance constrained by the shortest path through the network, otherwise, the euclidean distance will be calculated directly to the reference point.

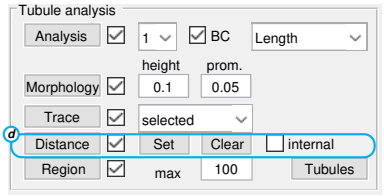


Figure 10.11: (d) Measurement of distance from a reference point

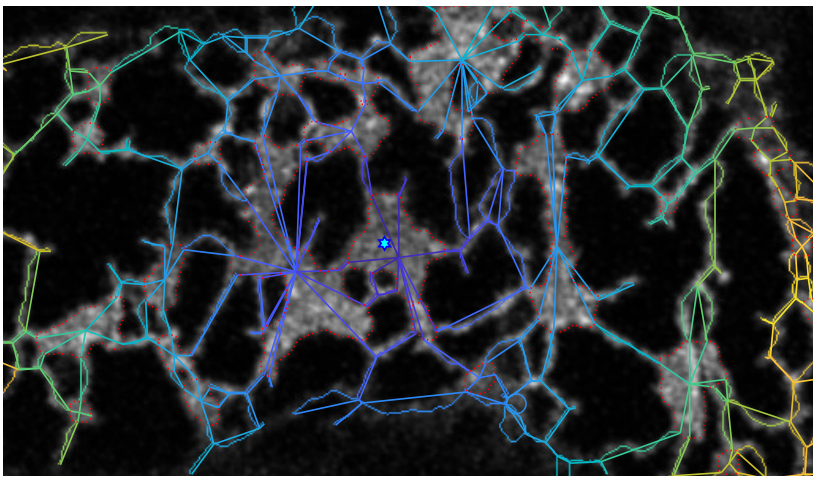


Figure 10.12: Euclidean distance measure from a manually-defined reference point

10.6 Local region measurements

The **Region** button will allocated any signal intensity within a maximum distance in pixels (set by **max**) from the tubules and cisternae to the nearest element of the skeleton (Fig. 10.13). This feature is being developed to provide a distance-dependent spatial co-localisation analysis of other labelled proteins or organelles near the ER.

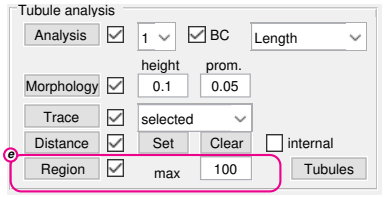


Figure 10.13: (e) Measurement of intensities in the region of the network

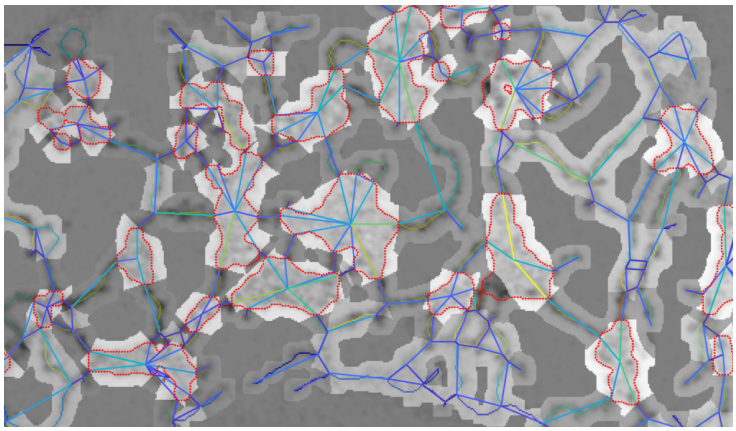
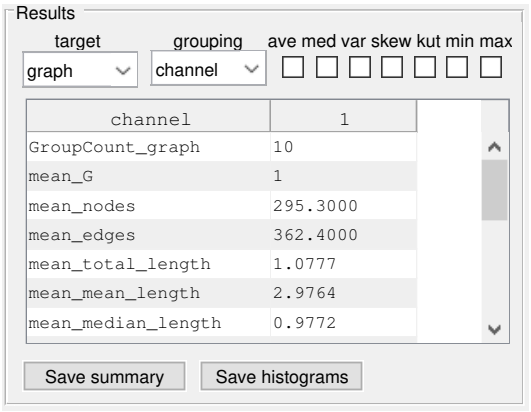


Figure 10.14: Merged image showing the local regions adjacent to each tubule or cisterna. Signal from these regions is re-allocated to the nearest element

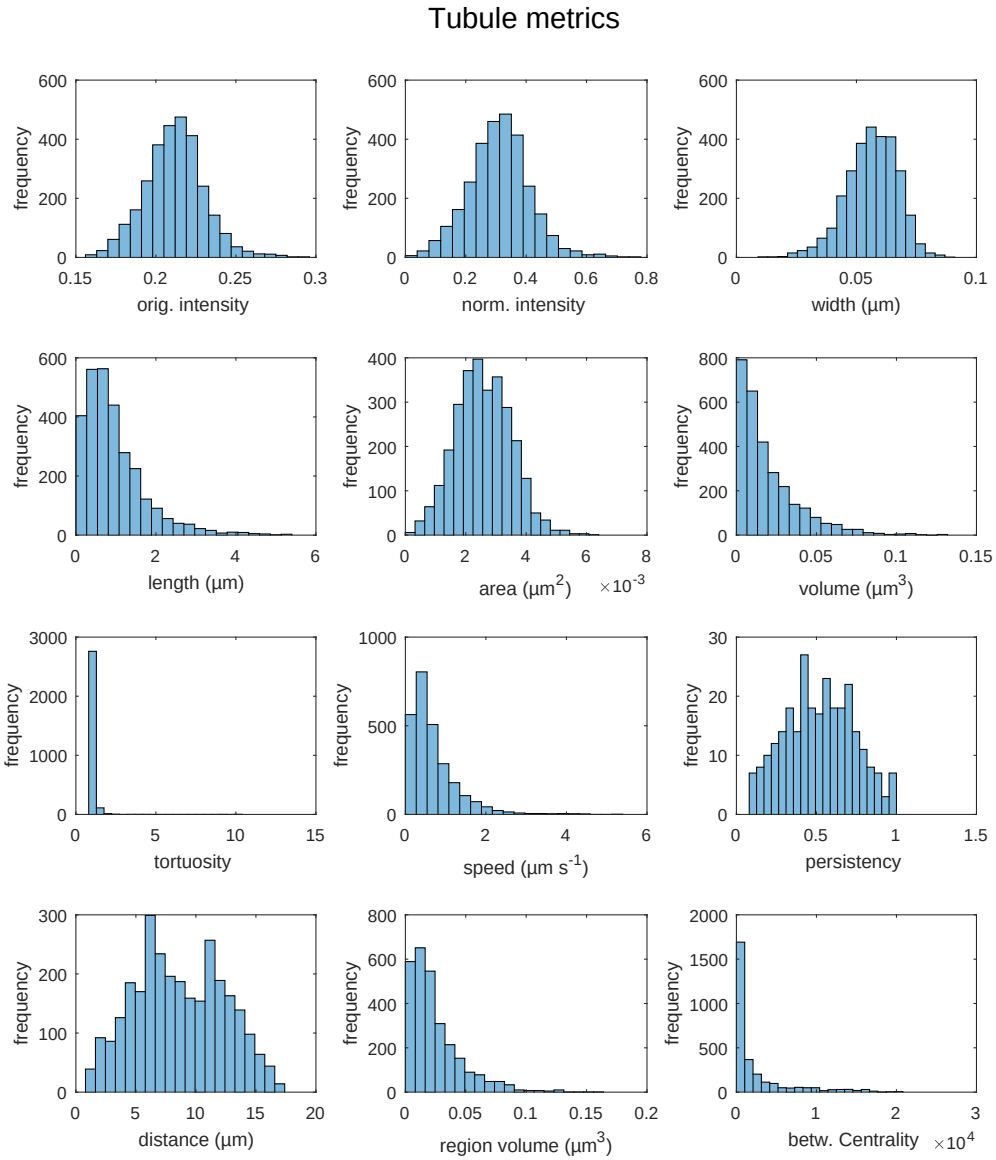
10.7 Data display

Results are displayed in tabular form in the **Results** panel (Fig. 10.7).



The **target** dropdown menu is used to select which results are shown from *tubules*, *nodes*, *cisternae*, *polygons*, *morphology*, *perimeter*, *trace* or textitgraph, whilst the **grouping** dropdown menu can be used to aggregate the data by just the *current* image, or by *channel*, *section*, *frame* or *all*. Normally results are summarised as the mean for the level of grouping chosen. However, additional metrics can be selected using the adjacent checkboxes, to include higher moments of the distribution inclding *median*, *variance*, *skewness*, *kurtosis*, *min* and *max*. These are added to the tabulated results and also included when the data is saved.

The **Save histograms** button generates a histogram for each metric and category and presents these in separate windows that can be saved or printed (Fig. 10.7).


Tubule metrics
frequency
orig. intensity
norm. intensity
width (µm)
length (µm)
area (µm^2) ×10^{-3}
volume (µm^3)
tortuosity
speed (µm s^{-1})
persistency
distance (µm)
region volume (µm^3)
betw. Centrality ×10^4

10.8 Data output

The data and images from the analysis can be saved using the controls in the **Output** panel (Fig. 10.15).

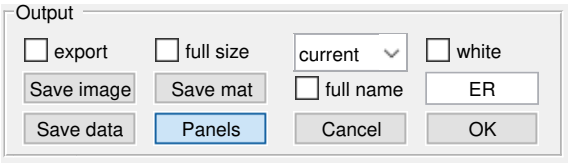


Figure 10.15: Output controls used to save data and images

The **Save image** button prompts the user for a filename and saves a version of the image and colorbar currently displayed with all the annotations in '*.png' and '*.pdf' format. If the **full size** checkbox is activated, the images will be saved in their native resolution. The *current* dropdown menu controls the background for some graph plots, and can be set to white or black to aid visibility. Likewise, the *white* checkbox forces the background to white for some plots.

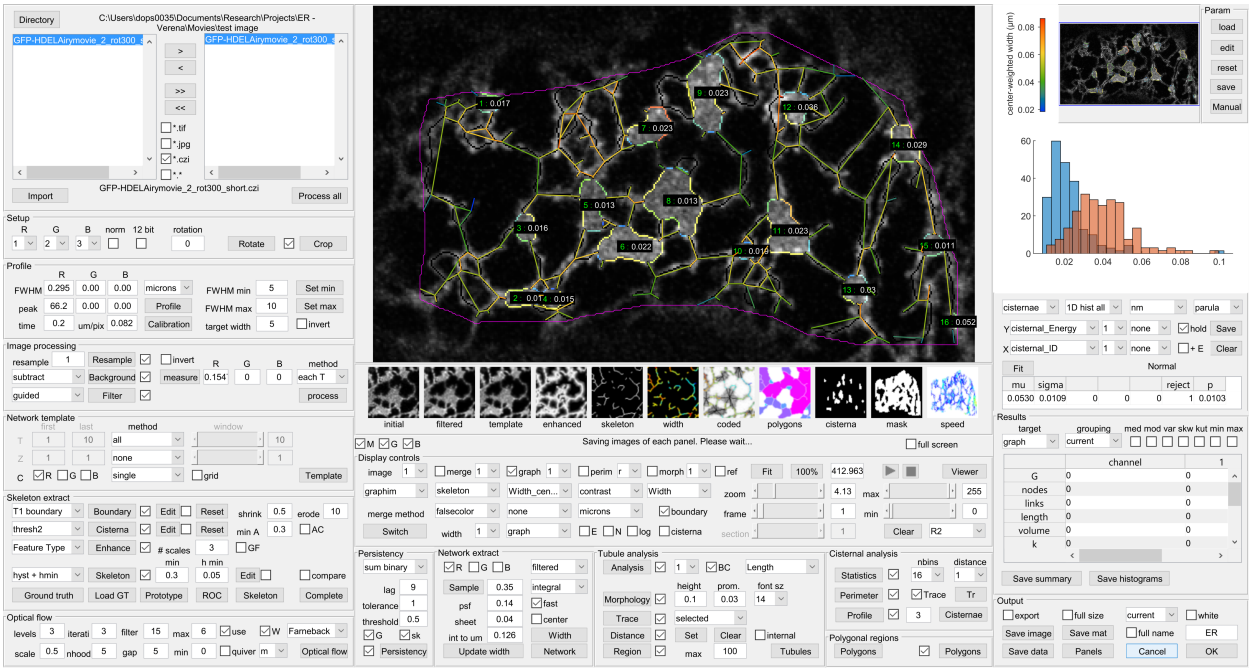
If the **full name** checkbox is ticked, the filename entered will be affixed with the original filename. Alternatively, if left unchecked, the **prefix** will be appended instead.

The **Save data** button writes all the data on the ER tubules, ER cisternae, tubule morphology and polygonal regions to separate sheets in an Excel file. Each sheet includes columns specifying the *filename, channel, section* and *frame*. Note: if any metric includes a NaN or Inf value, it appears to be replaced in Excel with a value of 65535.

The **Panels** button saves a copy of the all the panels in the interface as '*.png' files that can be used to tailor the illustrations in this manual to any specific application. The **export** checkbox will cause images to be saved at each step in the entire processing sequence. This is useful to generate a record of the analysis, but is slow as many large images will be written to disk. The **white** checkbox and adjacent dropdown menu, determine the background image when automatically saving some of the graphs.

11

Analysis of the ER cisternal structure



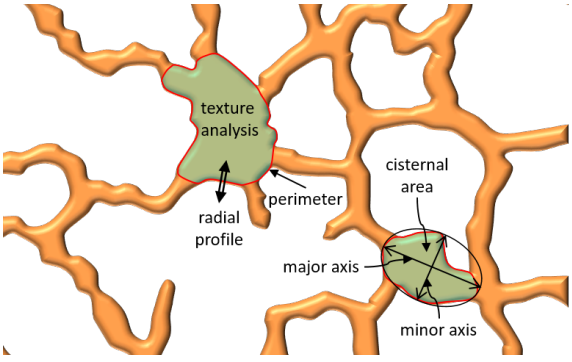


Figure 11.1: Schematic diagram of ER cisternae illustrating the different types of morphological and texture based analysis that are possible

Once the ER cisternae have been segmented (see Chapter 9), the **Statistics** button in the **Cisternal analysis** panel (Fig. 11.2(a)) calculates various morphological metrics for each binary object at each time-point (Table 11.1), that collectively describe the shape, size, and texture of each cisterna.

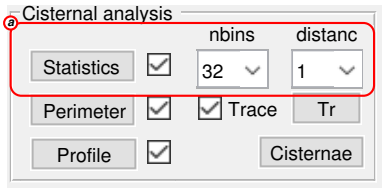


Figure 11.2: Schematic outline of ER cisternal analysis

Metric	Explanatory notes
ID	The node identity displayed on the image overlay
area	The area of each cisterna
major axis and *minor axis*	length (in pixels) of the major and minor axis of the ellipse that has the same normalized second central moments as the cisterna
orientation	The angle between the x-axis and the major axis
solidity	The proportion of the pixels in the convex hull that are also in the cisterna
perimeter	The perimeter of the cisterna
elongation	The ratio of the major axis to the the minor axis
roughness	The ratio of the perimeter^2 to the area
circularity	The ratio of the radius determined from the area to the radius determined from the perimeter

Table 11.1: Morphological metrics calculated for each cisternal sheet

The cisternal image can also be used to extract intensity-based metrics from the background-subtracted image or indeed other images, such as the optical-flow speed or persistency (Table 11.2).

Metric	Explanatory notes
mean Intensity, *max Intensity* and *min Intensity*	metrics calculated for each cisterna from the background-subtracted image
node degree	The number of connecting tubules incident on the cisterna
node strength	The sum of the widths of the tubules connecting to the cisterna (μm)
average node strength	The *node strength* divided by the *node degree*, (μm)
max and mean persistency	The maximum and average persistency of pixels in each cisterna (s)
speed - local, *speed - max* and *speed - global*	The scalar sum of local speeds at every pixel, the maximum speed across all pixels, and the global (vector sum) speed (μms^{-1})
Dir - local, *Dir - std*, and *Dir - global*	The mean angular direction of movement for all pixels in the cisterna along with the angular standard deviation, and the mean global drection from the vector sum (°)
Flow coherence	The ratio of the local speed (scalar sum) to the global speed (vector sum)
Flow curl, *std*	The local angular rotation of the vector field within the cisternae (°)
Flow AV, *std*	The local angular velocity of the vector field (°s^{-1})
Flow divergence, *std*	The divergence of the vector flow field (°)
Persistency, *cv*	The mean period of time that each pixel forms part of a tubule (s)
max distance and *mean distance*	The maximum and average distance of any pixel to the edge of the cisterna (μm)
variance	The square of the difference in intensity of each pixel from the mean within the cisterna

Table 11.2: Intensity-based metrics calculated for each cisternal sheet

11.1 Texture metrics

Texture analysis provides some information on whether the cisternae can be regarded as homogeneous sheets, or include additional structure that might be consistent with a bundle of closely ap-

pressed tubules[1] or heterogeneous protein distribution. An analysis of the *texture* within each cisternal sheet is calculated from the Gray-Level Co-occurrence Matrix (GLCM, Table 11.3)[2]. For each pixel in the segmented cisternal region, the GLCM examines pixels within a certain distance, set by the **distance** textbox, in four directions (*NW,N,NE,E*) and constructs an accumulator array where the row and column indices correspond to the intensity value of the target pixel and the neighbours, grouped into a number of equal intensity bins. The number of bins in the GLCM matrix is set by the **nbins** textbox, and a value of 32 gives reasonable intensity discrimination without the GLCM becoming too sparse. In the current implementation, the *distance* is measured in multiples of the radius of the smallest tubule, set by $FWHM_{min}$. The GLCM is normalised so the value at each location corresponds to a probability $p(i, j)$. Pressing the **Statistics** button will display a colour-coded GLCM (Fig. 11.3).

The *contrast, correlation, energy* and *homogeneity* of the GLCM are calculated individually for each cisterna (Table 11.3). If the value of distance is greater than 1, the GLCM is calculated and summed for each integer distance up to the value set. In addition, results are summarized for all the pixels in the network to give a single value for *contrast all, correlation all, energy all* and *homogeneity all*. These texture metrics are regarded as orthogonal descriptors of the GLCM, and represent a compact summary of the texture information (Table 11.3).

[1] C.J. Nixon-Abell, J.and Obara, A.V. Weigel, D. Li, W.R. Legant, C.S. Xu, H.A. Pasolli, K. Harvey, H.F. Hess, E. Betzig, C. Blackstone, and J. Lippincott-Schwartz. Increased spatiotemporal resolution reveals highly dynamic dense tubular matrices in the peripheral er. *Science*, 354:aaf3928, 2016

[2] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, pages 610–621, 1973

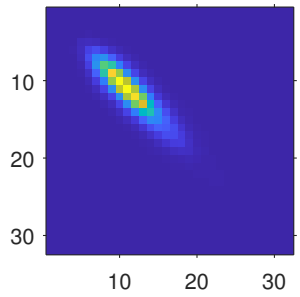


Figure 11.3: A Gray-level Co-occurence Matrix (GLCM)

Metric	**Explanatory notes**	Equation
contrast	A measure of the intensity contrast between pixel i and its neighbour j. Values range from 0 to (nbins-1)^2. Here, results are normalised to (nbins-1)^2 to fall between [0 1]. An idealised cisternal sheet would have a contrast of zero.	$\sum_{i,j} \lvert i-j \rvert^2 \, p(i,j)$
correlation	A measure of how correlated a pixel is to its neighbour. where μi and μj are the weighted mean intensities, and σ_i and σ_j are the standard deviations of the GLCM distributions. Values range from -1 (un-correlated) to 1 (fully correlated). An idealised cisternal sheet would have a value of 1.	$\sum_{i,j} \frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j}$
energy	Gives the sum of squared elements in the GLCM. Values range from 0 to 1. An idealised cisternal sheet would have an energy of 0.	$\sum_{i,j} p(i,j)^2$
homogeneity	Measures the closeness of the distribution of elements in the GLCM to the diagonal. Values range from 0 to 1. An idealised cisternal sheet would have a value of 1	$\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j \rvert}$

Table 11.3: Metrics calculated from the Gray-level Co-occurrence Matrix

11.2 *Cisternal perimeter measurements*

The **Perimeter** button traces the pixel intensity around the perimeter of each cisterna, excluding the junctions with the tubules (Fig. 11.4(b)).

The result can be visualised using the **graph** checkbox, with the dropdown menu below selecting one of the metrics beginning with **P** (Fig. 11.5). The relevant metric is plotted on as a graph or pixel skeleton (Fig. 11.6)

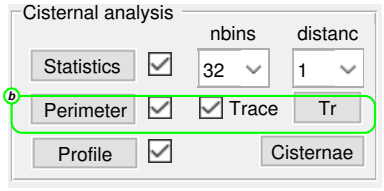


Figure 11.4: (b) Controls to measure along the perimeter of the cisternae

Figure 11.5: Controls to display the results of the cisternal analysis

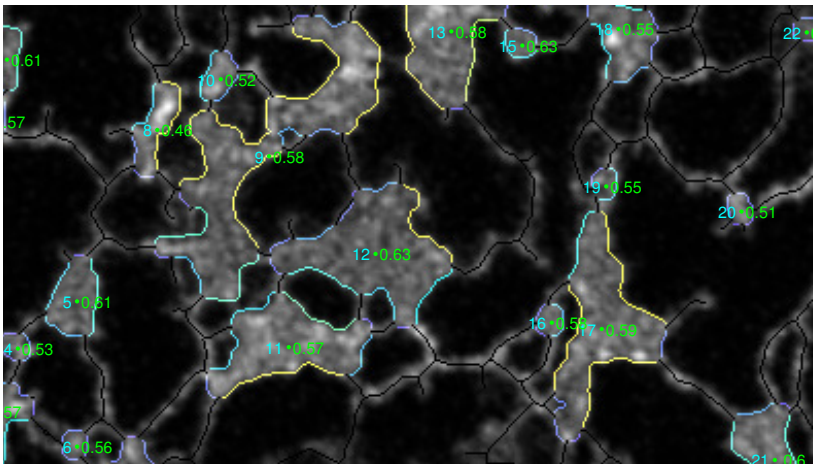


Figure 11.6: Cisternal perimeter color-coded by length, annotated with the cisternal ID (cyan) and homogeneity (green)

Perimeter profiles can also be presented unwrapped as a set of horizontal lines representing the intensity along the profile stacked to give an image if the **Trace** checkbox is active. The **Tr** button calculates the trace and displays it in a separate window that can be saved independently. The trace function is most useful in multi-channel images to see how the two spearate protein distributions relate to each other along the cisternal boundary.

11.3 *Cisternal profile measurements*

Rather than trace along the perimeter of the cisternae, the **Profile** button (Fig. 11.7(c)) measures the average intensity distribution along radial transects normal to the boundary of the cisterna both inwards and outwards (Fig. 11.8). this is most useful for multi-channel images where one labelled protein is localised to the cisternal perimeter, whilst the other is localised to the ER lumen, for example.

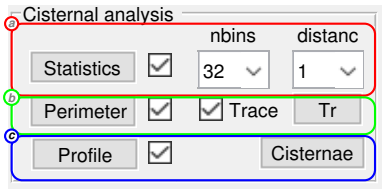


Figure 11.7: Controls to measure along the perimeter of the cisternae

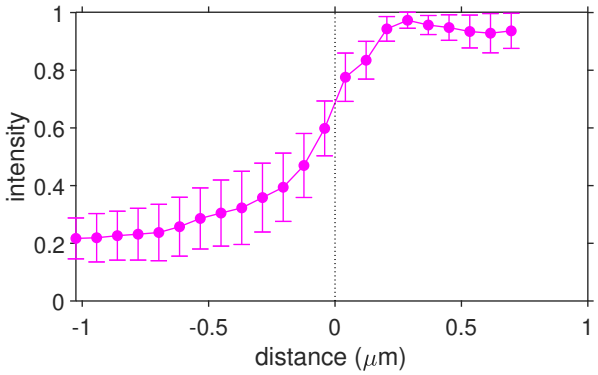


Figure 11.8: average radial transect

12
Analysis of the intermembrane polygonal regions

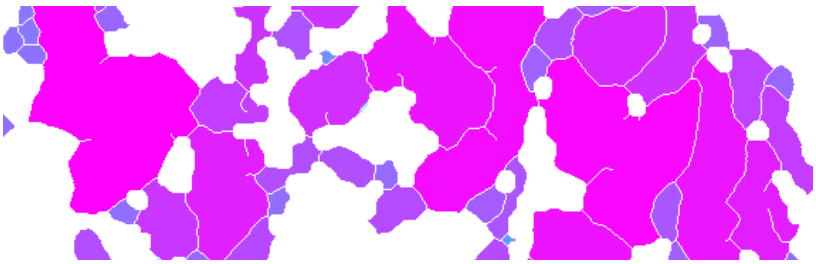


Figure 12.1: Segmentation of the polygonal regions enclosed by the ER network. Each region is colour-coded by it's area

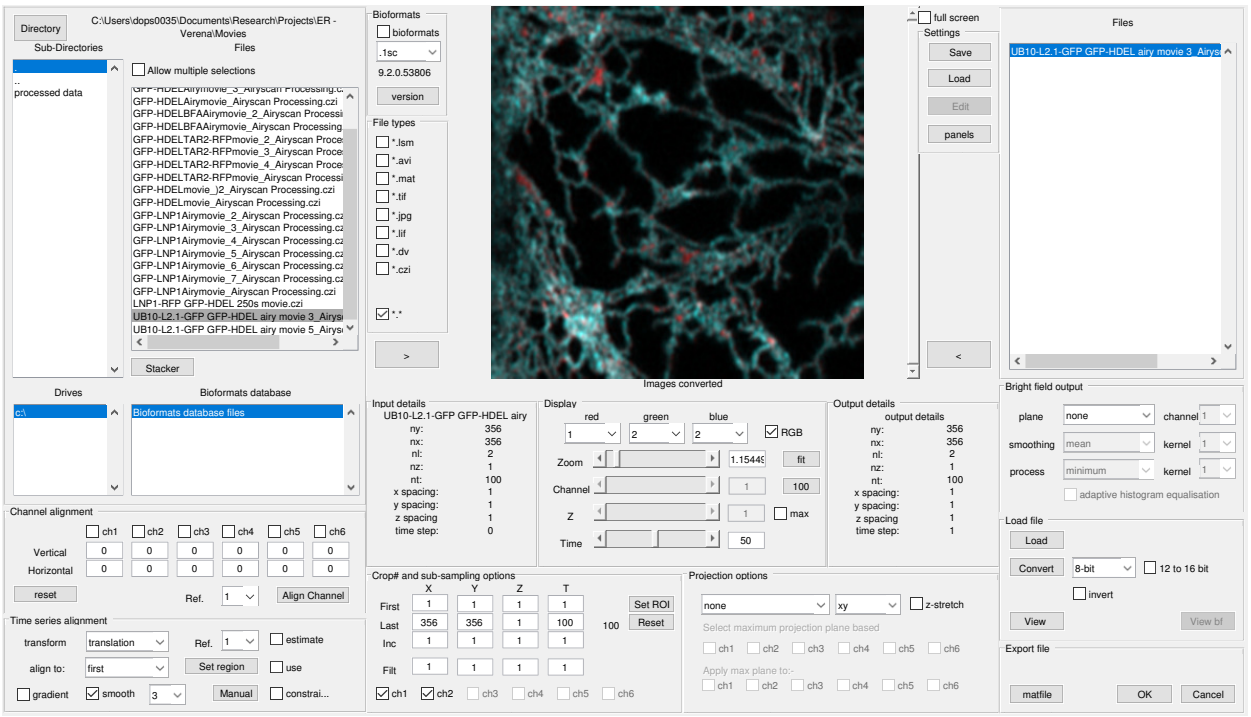
The **Polygons** button in the **Polygonal analysis** panel (Fig. 12.2) segments the polygonal regions that are fully enclosed by the ER network (Fig. 12.1) and calculates various morphological metrics at each time-point (Table 12.1) in a similar manner to the ER cisternae.

Figure 12.2: Controls for analysis of polygonal regions

Metric	**Explanatory notes**
ID	The identity of the polygonal region
area	The area of each polygonal region with the ER tubules thinned to a single-pixel wide skeleton
major axis and *minor axis*	length (in pixels) of the major and minor axis of the ellipse that has the same normalized second central moments as the region
orientation	The angle between the x-axis and the major axis
solidity	The proportion of the pixels in the convex hull that are also in the region
perimeter	The perimeter of the region
elongation	The ratio of the major axis to the the minor axis
roughness	The ratio of the perimeter^2 to the area
circularity	The ratio of the radius determined from the area to the radius determined from the perimeter
max distance	The furthest distance within the region to the ER network
mean distance	The average distance within the region to the ER network
exclusive area	The area excluding the width of the ER tubules

Table 12.1: Morphological metrics calculated for each cisternal sheet

13
Image Import



13.1 Introduction

This package was originally designed to import confocal image stacks into Matlab, particularly for ratio imaging applications. The import options allow some channel registration, time-series alignment, sub-sampling, smoothing and z-projection. Most file formats can now be handled using the Bio-Formats program (Linkert et al. 2010[1]) called by the import functions:

`http://www.openmicroscopy.org/site/support/bio-formats4/`
`http://loci.wisc.edu/software/bio-formats`

The import routines for Zeiss *.lsm images use the LSM file toolbox written by Peter Li and available on the Mathworks Website.

`http://www.mathworks.co.uk/matlabcentral/fileexchange/8412-lsm-file-toolbox`

[1] M. Linkert, C. T. Rueden, C. Allan, J. M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. Macdonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. W. Eliceiri, and J. R. Swedlow. Metadata matters: access to image data in the real world. *J Cell Biol*, 189:777–82, 2010

The latest versions of Java needs to be installed, and is available from:

http://www.java.com/en/

13.2 File Selection

The file selection panel shows:

- a **Directory** button that opens a standard dialog box to select a different directory
- sub-directories of the current folder
- individual files in the current directory
- the currently selected file
- the available drives
- image files within a Leica "lif" database (if appropriate)

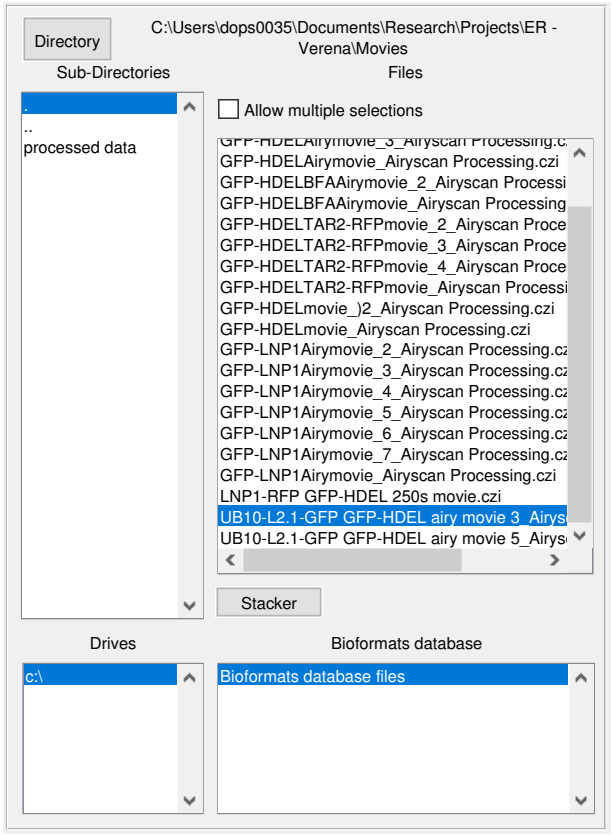


Figure 13.1: File selection panel

If the **Allow multiple selections** checkbox is ticked, a range of files can be loaded in one operation. This is useful to join a set of consecutive sequences from the same time-series experiment. The images must have the same dimensions in x and y, and have the same number of channels. Images are sorted in alphabetical order in the listbox and will be imported in this order. If a different order is required, each file has to be added in sequence manually.

A set of checkboxes (Fig. 13.2) are available to display a restricted set of file types or all files in the directory (*.*)

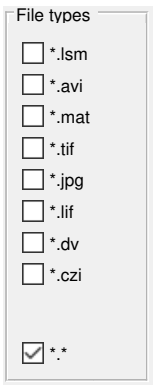


Figure 13.2: Setting the file extensions

Once the required file(s) have been selected, the information on each file can be loaded by double clicking or using the **right-arrow**.

If the system can read the file format information, The file name(s) will then appear in the **output list** box, and the image details shown in both the **Input details** and **Output details** panels (see Fig. 13.3). If multiple files have been selected, only the details of the last file will be shown. Input and output details will be the same at this stage as no additional processing steps have happened.

If the system can not pick the pixel size information or time interval, a dialog box will appear with prompts for the user to enter the required information (see Fig. 13.4).

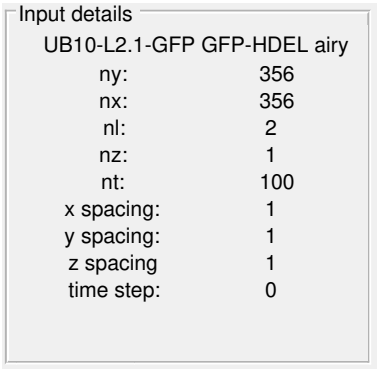


Figure 13.3: The input details panel: displays of the size and calibration values for the current image

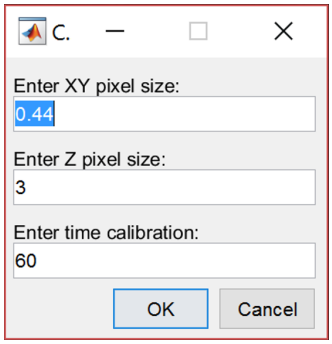


Figure 13.4: Dialog box prompting for the x,y,z spacing and the time-interval

13.3 *Combining separate channels*

The **Stacker** button is used to combine individual *.tif images from multiple channels and time-points into a single matlab file if they are numbered in sequence. The user is prompted for:

- the number of channels to combine and the pixel resolution (Fig. 13.5).
- an Excel file that contains the time stamp for each image as a single column. If this is not present, then the time interval defaults to 1;
- the first image for channel 1 (to get the core filename)
- the first image for channel 2 (to get the second core filename).

The images are then automatically loaded and concatenated into a single matlab file which is automatically saved. The matlab file can be loaded as normal.

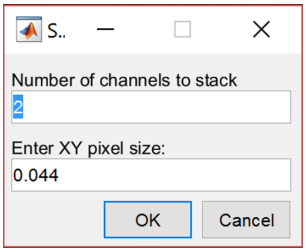


Figure 13.5: Setting the number of channels and pixel spacing when combining separate tif images

13.4 *Importing images from different formats (including Leica databases)*

The package can open most file formats using the Bio-Formats package. Leica images are stored in a single *.lif database and cannot be accessed directly. If a "lif" database is selected, or a format that cannot be read in directly, the Bio-Formats program

reads in the list of image stacks or time-series and displays these in a separate pop-up window (see Fig. 13.6).

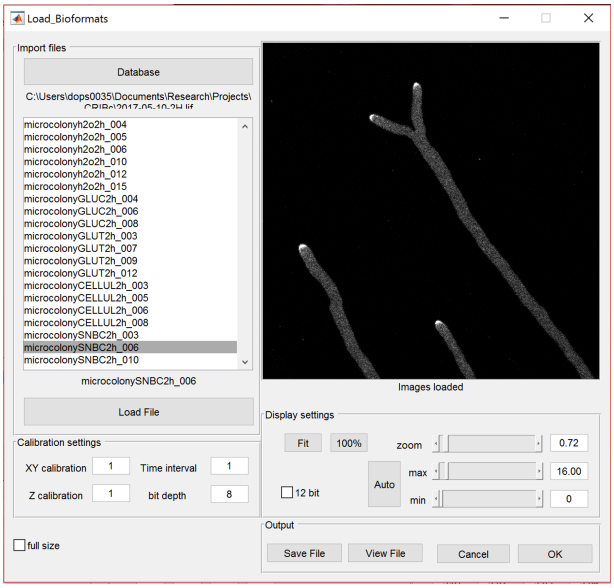


Figure 13.6: The Bio-Formats import window: Once a Leica database has been loaded, one of the image files can be selected and imported into the MatLab environment. The calibration parameters have to be set manually

The file is selected by double clicking in the list box and the name appears in the box underneath. Clicking the **Load file** button will import the image and display the first three channels of the first time/z-plane in the window. The **x**, **y** and **z** pixel sizes have to be entered manually into the appropriate boxes, along with the **Time** interval and **bit depth**.

Clicking **OK**, will import the selected file into the main interface. It is also possible to view the full files at this stage using the **View file** button, or save the file as a matlab array using the **Save file** button.

13.5 *Image display*

Once the images have been loaded, the first image of a time series will be displayed in the image window (Fig. 13.7).

If the image is a *z*-stack, the median plane will be displayed. If it is a multi-channel image, the median channel image will be displayed. If multiple separate image files have been loaded, a different file can be displayed by selecting the appropriate file from the output list box. A number of controls are available to alter how the image is displayed (Fig. 13.8).

For multi-channel images, different channels can be assigned to the **R**, **G** and **B** image planes using the drop-down menus to construct a RGB image, which will be displayed if the **RGB** checkbox is active.

The image size can be increased using the **Zoom** slider. If the image is larger than the display window, horizontal and vertical scroll bars will appear. The **fit** button maximises the size of the image to fit within the display window. The **100%** button displays

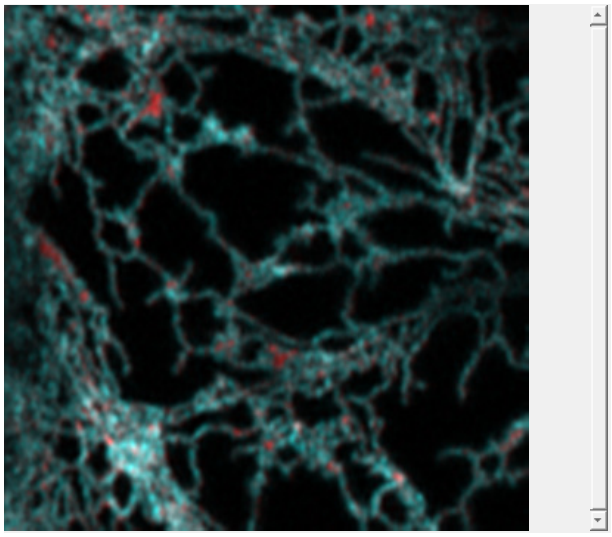


Figure 13.7: Image display panel: The image has been loaded in and an RGB version of the first three wavelength channels displayed. The red rectangle shows a user-defined region that will be cropped from the original file

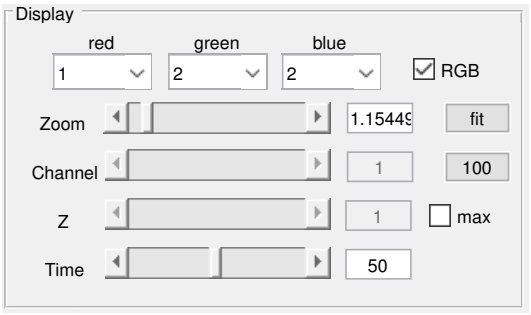


Figure 13.8: The display panel: Controls are provided to allow the user to select which channel, section or time-point to display, with further options to combine channels into an RGB image, or display a maximum projection of the z-stack (if present)

the image at a 1:1 image pixel to display pixel size.

The **Channel** slider displays a single channel image (if the RGB checkbox is un-ticked).

The **Z** slider scrolls through z-sections if the image contains 3-D (x,y,z) stacks. The **max** checkbox displays a maximum projection of the current z-stack.

The **Time** slider scrolls through each image in the time series.

13.6 Image crop and sub-sampling options

The image can be cropped by entering the **First** and **Last** pixel co-ordinates independently in the **X, Y, Z** or **T** text boxes (Fig. 13.9). Alternatively, a region-of-interest (ROI) can be selected in x and y using the **Set ROI** button. This prompts the user to draw a rectangular ROI on the image, which is then displayed in red. Completion of the ROI will update the values in the text boxes. Values can be reset using the **Reset** button if required.

The **Inc** text boxes allow (integer) sub-sampling independently in each image dimension.

The **Filt** text boxes allow spatial or temporal averaging over the designated number of pixels.

A series of checkboxes (**ch1**. . .**ch6**) are available to select which channels are to be included in the output image. Usually the bright-

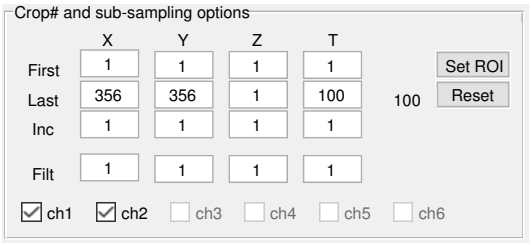


Figure 13.9: The crop panel: Provides controls to crop, sub-sample and filter images

field channel is not included in this selection, as it is processed separately (see Section 13.10 - Bright-field image processing).

The file details in the **Output details** panel should update to reflect the cropping and sub-sampling chosen. If multiple images have been loaded, the crop and sub-sampling options apply to all the files.

13.7 Alignment options between wavelength images

Options are available to correct slight mis-registration in *x,y* between individual wavelength images using the **Channel alignment** controls (Fig. 13.10). A reference channel, typically containing the best image selected using the **Z** and **T** sliders, is chosen using the **Ref channel** drop-down list.

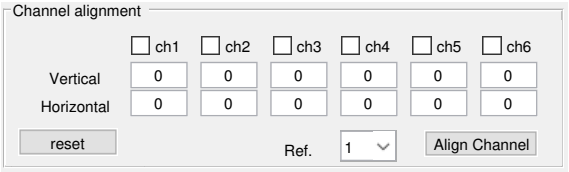


Figure 13.10: The channel alignment panel: Provides controls to allow sub-pixel registration between each channel and a selected reference channel

This acts as a template for cross correlation of any other channel, selected using the **ch1**. . .**ch6** checkboxes. The **Align Channel** button calculates the **Vertical** and **Horizontal** pixel offsets between the reference channel and the selected channels using cross-correlation across the whole image. These offsets are applied using bi-linear interpolation when the images are actually loaded.

The image display is modified during this process to display the before and after images in magenta and green.

13.8 Alignment options over time

Some level of correction for stage *x,y* drift or specimen movement can be achieved using the **Time series alignment** controls (Fig. 13.11). A reference channel (**Ref.**) is selected that has good contrast and features that are present in all images in the series, using the **Z** and **T** sliders, and, if necessary, a particular file if multiple files have been loaded simultaneously.

The **Set region** button prompts the user to select a ROI on the target image that will be used as a template to calculate the *x,y* pixel offsets, rotation and scaling for the corresponding image

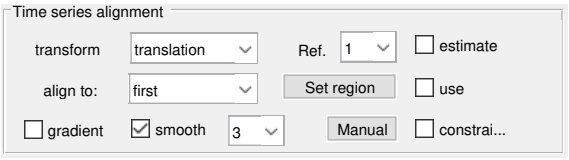


Figure 13.11: The Time series alignment panel: Provides controls to define a region for image alignment through the time-series, using a reference image

in subsequent time-points using cross-correlation and bi-linear interpolation. The region used as a template must be within the red ROI outline, if this has been used to crop the image, and is highlighted in blue. The same offsets are applied to all channels and **z**-planes for each time point. The alignment only takes place when the images are actually loaded if the **use** checkbox is ticked.

A number of different transform types can be applied with increasing degrees of freedom, including *translation*, for (*x*,*y*) translation, *rigid*, for translation and rotation, *similarity* for translation, rotation, and scale and *affine*, for translation, rotation, scale, and shear. The **estimate** checkbox can improve registration for some time-series by providing an initial estimate of the transform using phase correlation.

In addition, registration can be achieved by matching Speeded-Up Robust Features (**SURF**) or maximally stable extremal regions (**MSER**) detected in each image. These are much faster than registration using cross-correlation and can accommodate translation, scale and rotation, but are also sensitive to relatively small changes in key features due to brownian motion of vesicles for example.

The **align to:** drop-down menu sets the reference image to the first in the series or the previous image. In addition, the **smooth** checkbox allows smoothing with a kernel set by the drop-down menu prior to registration. Note: this smoothing only applies to calculation of the alignment transfom, not the actual image data. The **gradient** checkbox can be used to align based on the local image gradients, which may be useful if there are large homogeneous features in the image.

13.9 Image projection options

It is possible to reduce the image dimensionality by projecting data along the *z*-axis using the drop-down menu in the **Projection options** panel (Fig: 13.12).

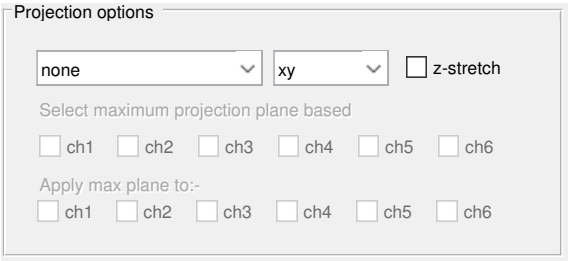


Figure 13.12: The projection panel: allows the user to reduce the image dimensionality by extracting a sub-set of the data in the z-dimension, or to re-orient the image stack to display xz or yz views. A number of different projection options can be selected and applied to different combinations of channels

- **Maximum:** displays the pixel with the maximum intensity in *z*

for each channel (maximum intensity projection or MIP). This is a common approach to visualise data, but should not be used as a precursor to quantitative measurements, particularly when ratioing two channels, as it selects the 'noisiest' pixel at the extreme of the distribution along the z-axis, and pixels from different positions in z for different channels will appear in the projected image, making a nonsense of the ratio image.

- **Minimum:** displays the pixel with the minimum intensity in z for each channel. This is rarely useful for fluorescence images, but can be helpful for bright-field processing. Nevertheless, it is recommended that bright-field processing is handles separately (see Section 13.10 - Bright-field image processing)
- **Average:** gives an average brightness projection in z, which provides good noise reduction and may be useful for simple objects that do not overlap in the z-direction.
- **Max plane:** this gives the user the option of selecting the z-position of the brightest pixel in one-or-more channels and then extracting the same (x,y,z) pixel (voxel) from the other channels. It is recommended that the imaged is smoothed in z (as well as x and y), before this operation to ensure that the brightest pixel is more likely to correspond to the centre of the object of interest. This approach is required if different channels are going to be ratioed later on to ensure that information from the same (averaged) voxels are compared.
- **Mx + mx plane:** This provides an option to calculate the maximum plane projection, based on specific selected channel(s) that extracts the appropriate (x,y,z) voxel from a second set of channels, selected by the second set of checkboxes. The remaining channels are processed using a simple maximum. This is useful for quantitative ratioing for the max plane images, which typically involve two wavelength channels and an autofluorescence channels for bleed through correction, and a morphological representation of the other channels from the (smoothed) maximum intensity projection.

13.10 Bright-field image processing

A bright-field image is often collected simultaneously with the fluorescence channels using a (non-confocal) transmission detector. Bright-field images can be processed separately to accentuate more useful information using the **Bright field output** controls (Fig: 13.13). The simplest form of processing is a **single plane**, selected by the position of the Z-slider from the bright field channel (**channel**). An amount of noise filtering can be applied using the process selected by the **smoothing** drop-down list (*mean*, *median* or *Wiener*) over a square x,y region specified by the **kernel** drop-down list.

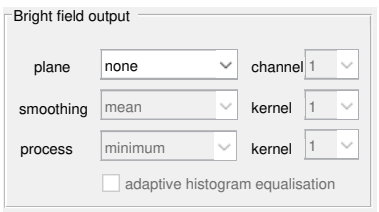


Figure 13.13: The bright field output panel: Provides controls to allow selection of single bright-field image planes or various algorithms to project the bright-field images, along with some contrast enhancement

If a projection option is chosen, the process required can be selected from the **process** drop-down menu. The algorithms available are designed to highlight pixels that might contain the most useful information within a local neighbourhood defined by the **kernel** drop-down menu.

Whether a projection of single plane option is chosen, the contrast of the resulting image can be improved using contrast-limited adaptive histogram equalisation (CLAHE) by checking the **adaptive histogram equalisation** box.

13.11 *Saving or loading the processing settings*

The **Save settings** button in the **Save options panel** saves the settings used for processing in a matlab file format, with the filename and a *settings* suffix (Fig: 13.14. The parameters can be re-loaded using the **Load settings** at a later stage. Note: when the settings are re-loaded, the user is prompted to re-set the alignment box for the time-series, and the cropping ROI is not available and must therefore be set-up again.

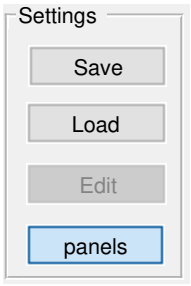


Figure 13.14: The Settings options panel: allows the user to save the loaded image and the processing settings

The **Save mat** button saves the processed fluorescence images and, if appropriate, bright-field images in a matlab file format. This can be re-loaded using the same interface at a later stage.

The **Save panels** button saves "png" images of each of the panels in the image and can be used to update this manual for any specific applications.

13.12 *Loading the selected files*

Once all the processing steps have been completed, the selected files can be loaded using the **Load** button (Fig: 13.15). If this is successful, the **View** button will be enabled and, if a separate bright field image has been processed, the **View bf** button. Clicking the **View** button will open a separate window with a video **Viewer** (see Chapter 14 - Viewer Program). There is an additional option to **Convert** the image format to *8-bit*, *16-bit*, *single* or *double* precision, using the adjacent drop-down menu. Images in 12-bit can be re-normalised to 16-bit using the **12 to 16 bit** checkbox. Images can also be inverted at this point using the **invert** checkbox.

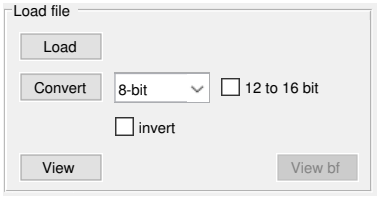


Figure 13.15: The Load file panel: The Load button imports the selected image(s) and applies all the alignment, sampling, smoothing and projection parameters chosen

13.13 *Saving the processed files*

If the image files have been loaded satisfactorily, clicking the **OK** button will return to the main program. The **Cancel** button will return to the main program, but without exporting the processed image file. The **matfile** button saves a copy of the processed image and information on the processing steps to a MATLAB file. This can be loaded later without having to re-process the original image.

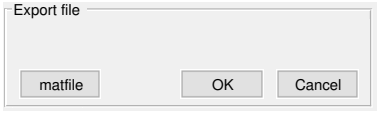
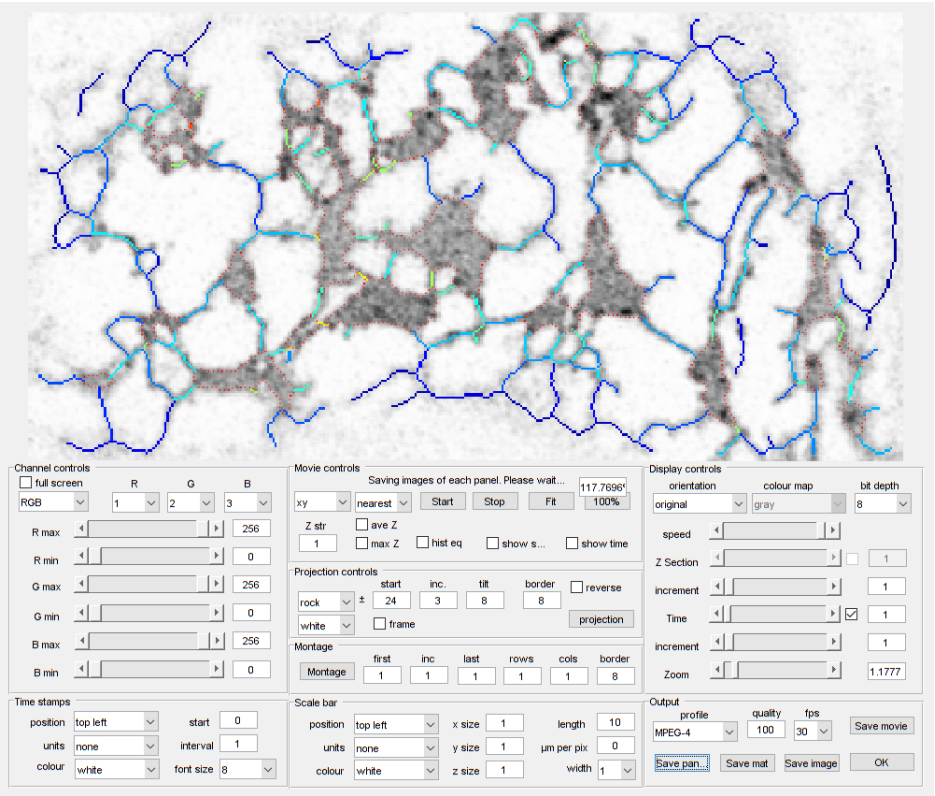


Figure 13.16: The export file panel to exit the program or save the processed image

14
Viewer Program



14.1 Introduction

The video **Viewer** allows the output of any processing steps to be animated on screen or saved in standard movie formats format. Images can contain multiple channels, although a maximum of three can be shown in RGB format. The first image in the file is displayed when the viewer opens. If the image is too large for the display window, the central portion will be displayed with scroll bars to move around the image.

14.2 *Movie playback controls*

The **Movie controls** (Fig. 14.1) can be used to **Start** or **Stop** playback. The normal image orientation is viewed in the *xy* plane. If the data has multiple *z*-sections, the orientation can be changed to view animations in *xz* or *yz* orientation using the drop-down menu. The **Z str** textbox allows control of the amount of interpolation in the *z*-axis needed to correct for the asymmetry in *x,y* and *x* pixel spacing. The type of interpolation can be set using the drop-down menu from *nearest*, *linear* or *bicubic*.

Figure 14.1: The movie controls panel: provides options to start/stop the animation or display max or average z-projections

The average (**ave Z**) or maximum (**max Z**) checkboxes generate the corresponding projection for the current time-point if the data has multiple *z*-sections.

The **hist eq** checkbox uses contrast-limited histogram equalisation (CLAHE) to improve the overall contrast in the image. If the data has more than one *z*-section, the maximum projection is calculated first before the CLAHE enhancement. The **show scale** and **show time** checkboxes toggle display of the time stamps set in the **Time stamps** panel, and the scale bar set in the **Scale bar** panel.

14.3 *Display controls*

The **Display controls** panel (Fig. 14.2) can be used to display images in their **original** orientation, or as a rotated image (*90 left*, *90 right* or *180*) using the drop-down menu.

If the image is initially a single intensity channel the **colour map** drop-down menu can be used to apply a colour look-up table (LUT) to the image.

The **bit depth** controls the number of bits to scale each channel in the display, and automatically adjusts the maximum scaling of the channel intensity controls.

Note, many microscopy images contain 12 bits of information (0-4095), but are stored in 16-bit format. This means the bit-depth may be automatically set to 16 (0-65535) when the image is loaded. As a result the screen will appear black until the bit-depth is set to 12. In other situations, the software cannot pick up the correct setting for the bit-depth and defaults to a value of 1. Images may appear completely saturated until the bit-depth is set to the appropriate value, usually 8.

The **speed** slider adjusts the speed of playback.

The **Z Section** or **Time** sliders select a specific *z*-section and time point and also have **check-boxes** to control whether images are

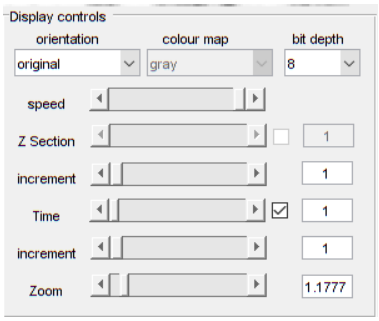


Figure 14.2: The display controls panel: set the image orientation and control the dimensions for animation

animated in this dimension during playback.

The **increment** slider controls the gap between images during playback.

The **Zoom** slider controls the scale of the image displayed. If the image is larger than the display window, horizontal and vertical scroll bars appear automatically.

14.4 *Image brightness controls*

Images are usually imported in RGB format where each channel corresponds to a different wavelength image. However, pseudo-colour coded ratio images are constructed in HSV colour space where the parameter of interest is coded as Hue, and the intensity and/or saturation are used to represent the strength of the original signals. Switching between RGB and HSV display is achieved with the **RGB** drop-down menu on the **Channel controls** panel (Fig. 14.3).

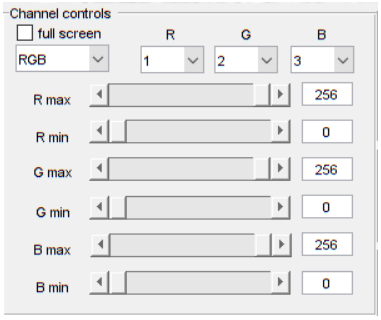


Figure 14.3: The channel controls: Select the channels to display and the minimum and maximum intensity for each channel

In RGB mode, the **R, G** and **B** drop-down menus can be used to select which channel should be displayed in which colour plane, and the sliders used to set the **minimum** and **maximum** values for each colour independently.

In **HSV** mode, the slider and drop-down labels change to represent the **Hue**, **Saturation** and intensity **Value**. This allows independent scaling of these parameters (but only yields a useful image if the data is in HSV format).

14.5 *Projection controls*

If the image contains an RGB z-stack or a series of z-stacks collected over time, the **Projection controls** panel (Fig. 14.4) can be used to construct a rocking or tilted animation of the maximum projection of each channel for each z-stack at different angles.

Figure 14.4: The projection controls: Allows calculation of tilting or rocking projections of z-stacks

The **start** text box sets the initial angle for the rotation, whilst the **inc.** text box controls the angle between projections. The **tilt** or **rock** is calculated symmetrically about zero between these limits.

The **Z str.** Sets a z-stretch to correct for the asymmetric sampling in x,y and z. This is calculated from the nominal z-pixel spacing divided by the x,y pixel spacing. However, additional correction may be required depending on the lens and immersion media using to collect the original data. Ideally, values are based on a calibrated sample, such as a 15 μm fluorescent sphere, to ensure correct geometric scaling.

If the original data is an HSV ratio image, the **HSV** setting needs

to be selected in the channel controls panel, and the projection will be calculated initially as a maximum projection of the intensity value (**V**). The *x,y,z*-pixel co-ordinate of this maximum brightness pixel is then used to extract the corresponding values from the Hue and Saturation channels. This approach allows projection of multi-dimensional pseudo-colour coded images.

14.6 Annotation controls

The **Time stamps** control panel (Fig. 14.5) allows the user to add a time-stamp to the display. The **position** controls the location of the time label (top left, top right, bottom left, bottom right), whilst the **start** and **interval** text boxes and **units** drop-down menu control the information to be displayed in a specific **font size** and **colour**.

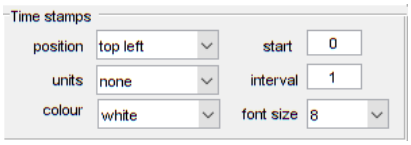


Figure 14.5: The annotation panels: allows addition of a time-stamp to each image

The **Scale bar** control panel Fig. 14.6 adds a scale bar to the image. The **position** controls the location of the scale bar, whose **length** (in microns) and **width** are set by the appropriate text boxes. The calibration between pixels and microns is set by the μ**m per pix** textbox.

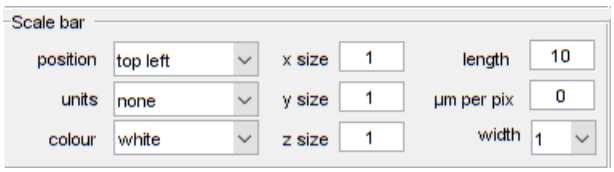


Figure 14.6: The annotation panels: allows addition of a scale bar to each image

14.7 Montage controls

A **Montage** of selected images (Fig. 14.7), defined by the **first**, **inc.** and **last** can be constructed in a matrix format defined by the **rows** and **cols** controls, with a gap between the images defined by the **border**. The montage appears in a new window and can be saved in a variety of formats.

Figure 14.7: The montage panel: Sets up a montage of images that can be subsequently saved in a grid format

14.8 Output controls

Processed images can be saved to a variety of video formats format using the **Movie** button in the **Output** panel (Fig. 14.8). This provides choice over the compression**profile** (default is MPEG-4), and,

if appropriate the **quality** and frames-per-second (**fps**) required. The image series can also be saved as a matlab array using the **Save mat** button, or as a single image using the **Save image** button.

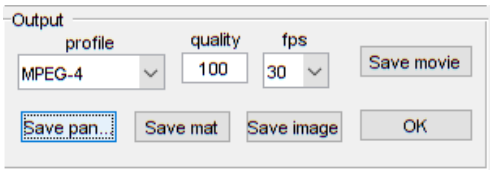


*Figure 14.8: The output panel: Saves the final movie in *.MPEG-4 format*

The **Save panels** button saves images of all the control panels to help in construction of an application specific manual.

15
Binary editing

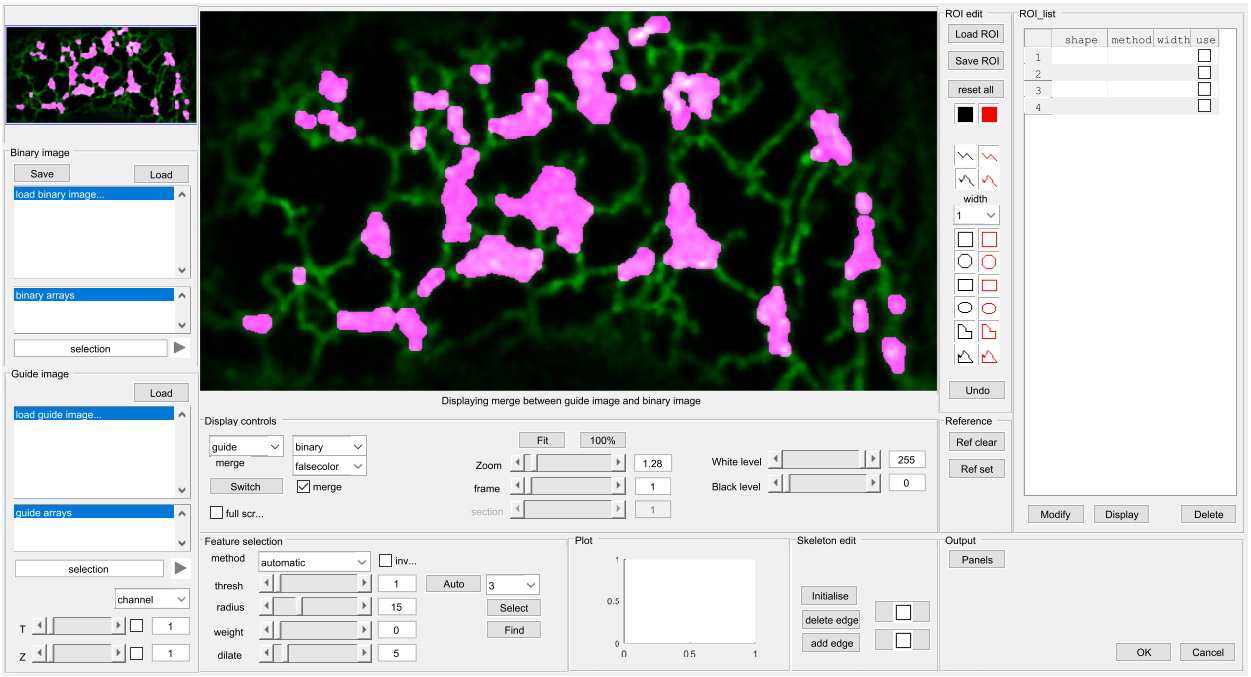


Figure 15.1: The GUI interface to edit binary images

15.1 Loading the images

Binary images are used to mask the boundary of the arena, delineate the food resources or other features, and represent the single-pixel wide skeleton. Whilst each of these can be generated automatically, it is often the case that they need to be edited manually. The *Binary editing* panel has tools to achieve this.

If the *Binary editing* window has been called from the network interface, by clicking the **edit** mask button for example, the template image and the current mask (if any) will automatically be displayed as a green-magenta merge when the window opens (Fig. 15.2).

The order of the merged image can be changed using the **Switch** button, the type of merge using the drop-down menu, or removed by un-checking the **merge** check-box (Fig. 15.3). The other display

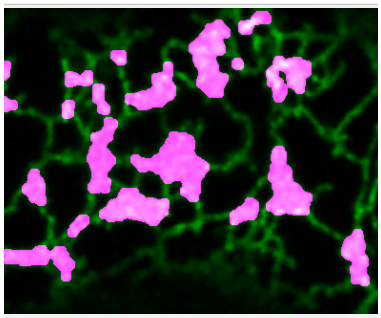


Figure 15.2: Main display window showing a merge between the guide image (green) and binary mask (magenta).

controls can be used to change the zoom, adjust the contrast, or, in multi-dimensional images, scroll through the sections and frames.

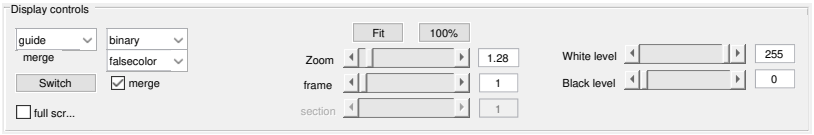


Figure 15.3: Main display window and controls to select the images to display, the type of merge, zoom and contrast.

In some instances, it may be desirable to load a previously constructed binary image, or a different guide image stored on disk. Available images (stored as MatLab arrays) can be loaded into the *Binary image* list-box (Fig. 15.4) using the **Load** button. The appropriate binary image from the arrays present in the matfile is selected by a single click, and displayed in the **selection** box. If the binary image is part of a structure within the matfile, it will be displayed in the **binary arrays** and has to be selected from this list-box. The adjacent right arrow is used to import it into the interface.

In a similar manner, a separate guide image can be loaded from a MatLab file on disk using the controls in the *Guide image* panel. If the guide image is part of a MatLab structure, the names of the available array are shown in the **guide arrays** list-box, and has to be selected from here.

The **channel** drop-down menu is used to choose the channel to import, whilst the **T** and **Z** sliders are used to select the frame and section, respectively. The check-boxes adjacent to the sliders give a maximum projection along that dimension. All changes are displayed in the main figure window. Once the most suitable combination has been chosen, the image is imported using the right arrow.

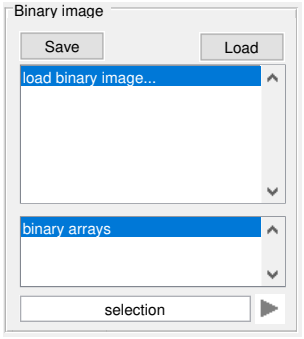


Figure 15.4: Controls to select and load a binary image (as a Matlab array).

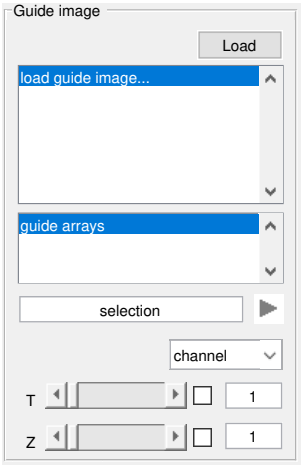


Figure 15.5: Controls to select and load a guide image (as a Matlab array).

15.2 *Automatic feature selection*

The *Feature selection* panel is used to try to automatically segment the features of interest (Fig. 15.6).

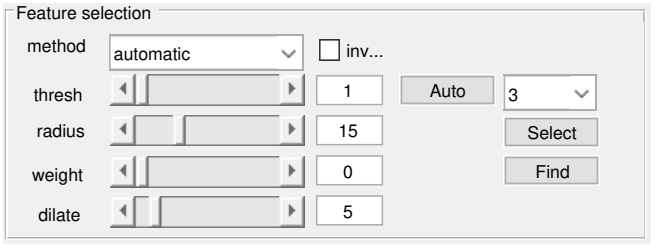


Figure 15.6: Controls to segment features in the image

The **method** drop-down menu provides access to a number of different thresholding strategies including:

- *manual:* The user chooses a threshold using the **threshold** slider.
- *automatic:* Automatically chooses a threshold when the **Auto** button is pressed using Otsu's criterion[1] for multiple thresholds,

[1] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, Cyber.*, 9:62–66, 1979

of which the lowest one is selected. The number of partitions is selected by the adjacent drop-down menu. A histogram of the image intensities is shown in the *plot* panel, with the position of the threshold(s) shown in green (Fig. 15.7).

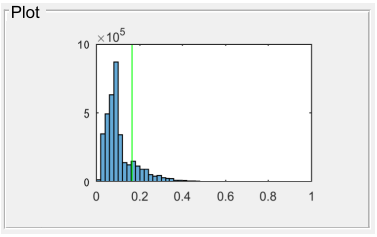


Figure 15.7: Histogram of piel intensities overlaid with the segmentation threshold

- *opening:* Applies a grey-scale opening to the image, with the radius determined by the **radius** slider. The image is segmented automatically using the lowest threshold from multi-level threshold menu;
- *tophat:* Applies a top-hat filter with the radius determined by the **radius** slider. The image is segmented automatically using the lowest threshold from multi-level threshold menu;
- *active contour:* Applies a grey-scale opening to the image, with the radius determined by the **radius** slider. The opened image is then matched to the underlying objects using an active contour algorithm.
- *local mean:* The local mean is calculated using a circular filter with the radius set by the **radius** slider and used as a local threshold. An additional constant can be subtracted from the threshold using the **weight** slider.
- *local median:* The local median is calculated using a circular filter with the radius set by the **radius** slider and used as a local threshold. An additional constant can be subtracted from the threshold using the **weight** slider.
- *midgrey:* The mid-grey threshold is calculated as the average of the local maximum and minimum, calculated using morphological closing and opening operations, respectively, with the radius set by the **radius** slider. An additional offset k can be set using the **weight** slider.

$$T(x,y) = \frac{max + min}{2} - k \tag{15.1}$$

- *Bernsen*[2]*:* The method calculates the local contrast and mid-grey values in a similar manner to the mid-grey algorithm from morphological operations. If the local contrast is above the contrast value k set by the **weight** slider, the threshold is set at the local mid-grey value. If the local contrast is below the contrast threshold, the pixel is classified as part of the object if the mid-grey value is above 0.5, or background if below.

[2] J. Bernsen. Dynamic thresholding of grey-level images. In *International conference on pattern recognition*, volume 2, pages 1251–1255, 1986

- *Niblack*[3]*:* The local threshold ($T(x,y)$) is calculated from the local mean ($m(x,y)$) and the local standard deviation ($s(x,y)$), with the radius set by the **radius** slider, weighted by a factor (k) set using the **weight** slider.

[3] W. Niblack. *An Introduction to Image Processing*. Prentice-Hall, 1986

$$T(x,y) = m(x,y) + k * s(x,y) \tag{15.2}$$

where k is typically -0.2.

- *Sauvola*[4]*:* The local threshold ($T(x,y)$) is calculated in a similar manner to the Niblack algorithm from the local mean ($m(x,y)$) and local standard deviation ($s(x,y)$), with the radius set by the **radius** slider, but the weighting is calculated as:

$$T(x,y) = m(x,y)\left[1 + k\left(\frac{s(x,y)}{R} - 1\right)\right] \tag{15.3}$$

where $R = max(s)$ and k takes small positive values, typically in the range 0.2- 0.5, set by the **weight** slider.

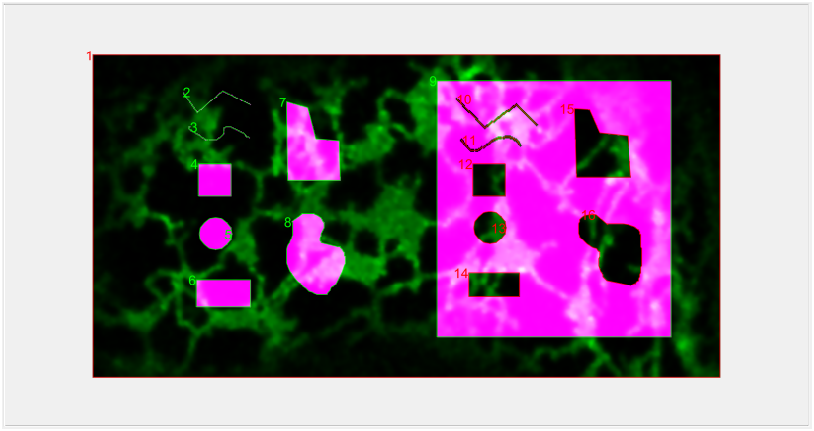
[4] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33:225–236, 2000

15.3 Manual Editing

If automatic segmentation is not possible, or the results need further modification, The *ROI edit* controls can be used (Fig. 15.8). Each icon depicts the type of ROI that can be drawn in black, or erased in red. The sequence of ROIs can be saved (**Save ROI)**) and reloaded (Load ROI). In addition, any binary image created with the *Binary edit* called from another program is imported with the existing list of ROIs.

The **reset all** button clears all the ROIs and the segmented image. The solid black icon set the ROI to fill the image, whilst the solid red icon erases the full screen. These should only be used as the first ROI in the series. Each ROI is numbered and listed in the adjacent *ROI list* panel (Fig. 15.9), which indicates the *shape* of the ROI, the *method* (*draw* or *erase*), the line width (only relevant for polylines and freehand lines), and whether the ROI should be used (**use** check-box).

Examples of the different ROI shapes are shown in Fig. 15.3.



For all the geometric shapes, a vertex is added with each left mouse click, whilst a double-click closes the shape. each vertex can be re-positioned by hovering over the vertex until a black circle appears and then dragging the circle to the new position. Additional vertices can be added along the lines using the **A** button on the keyboard. Once the shape is complete, it is plotted in green for an included region and red for an erased region, along with

Figure 15.8: Manual editing controls

ROI_list

	shape	method	width	use
1	full size	erase	1	☑
2	line	draw	0	☑
3	freehand	draw	0	☑
4	square	draw	1	☑
5	circle	draw	0	☑
6	rectangle	draw	0	☑
7	polygon	draw	0	☑
8	freehand	draw	0	☑
9	rectangle	draw	0	☑
10	line	erase	1	☑
11	freehand	erase	1	☑
12	square	erase	1	☑
13	circle	erase	1	☑
14	rectangle	erase	1	☑
15	polygon	erase	1	☑
16	freehand	erase	1	☑

Figure 15.9: List of current ROIs

a unique label matching the row in the *ROI list*. In addition, the corresponding binary image is shown overlaid in magenta. For the polyline and freehand line buttons, the line width can be chosen in pixels from the drop-down menu, or adjusted in the *ROI list* box.

ROIs can be altered or deleted using the **Modify** button or **Delete** button, respectively, in the *ROI list* panel (Fig. 15.10). The **Display** button can be used to show the ROI overlay and binary image at any time.

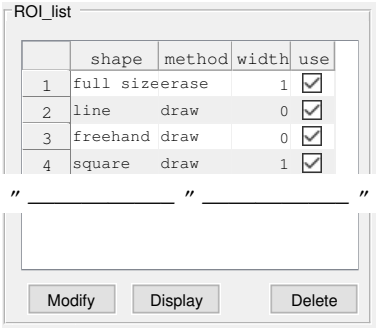


Figure 15.10: Modify, Display or Delete specific ROIs

15.4 Output

The **OK** button returns the adjusted binary iage to the calling program, along with the list of ROIs and their vertex co-ordinates. The **Cancel** button exits without saving any information. The **Panel** button saves a copy of each panel in *.png and *.pdf format for inclusion in the manual.

In addition, the binary image can be saved independently as a matlab file using the **Save** button in the *Load binary* panel (Fig. ref).

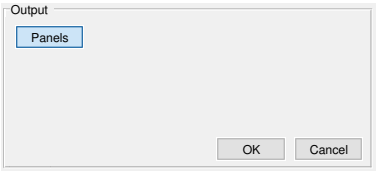


Figure 15.11: Output controls

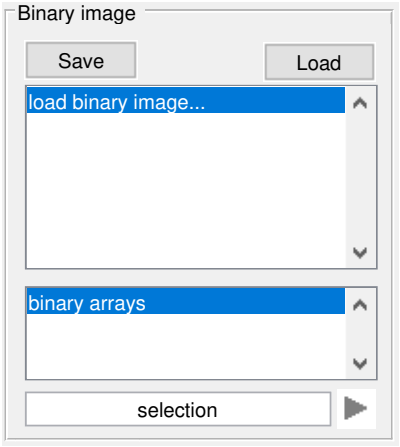


Figure 15.12: Option to save the adjusted binary image as a matfile

16
Parameter selector

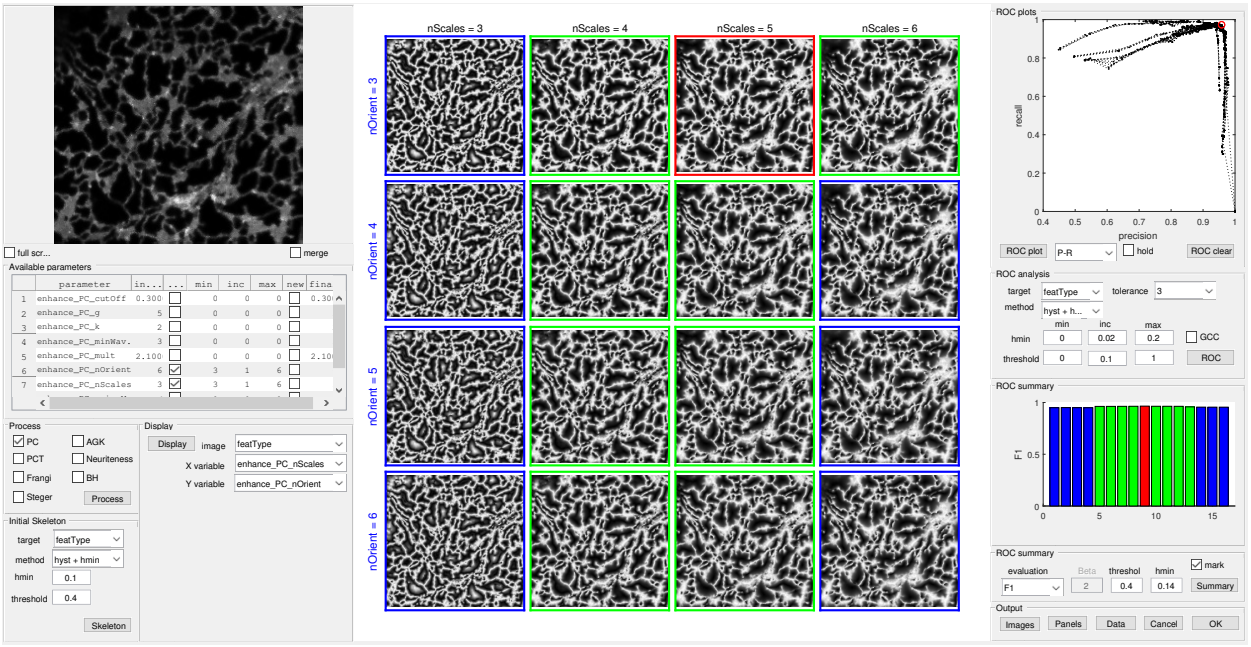


Figure 16.1: The GUI interface for the parameter selector program

16.1 Sensitivity analysis with factorial parameter combinations

The parameter selector panel allows a factorial combination of parameters to be applied to a region-of-interest (ROI) cropped from the image to compare with a manually-defined ground-truth. The program is called from the **Prototype** button in the *Skeleton extract* panel of the main interface (Fig. 16.2(a)).

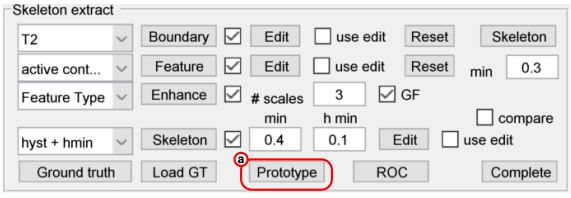


*Figure 16.2: The **Prototype** button in the skeleton extract panel*

The algorithm to apply to the ROI is chosen using the check-boxes in the *Process* panel. At the moment, only one algorithm

should be selected at a time. In a later version, multiple choices are planned to aid comparison between methods. Once a method has been selected, the parameter options specific to that method are displayed in the *Available parameters* panel (Fig. 16.4).

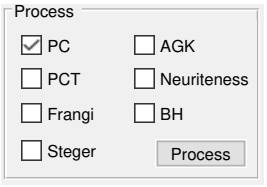
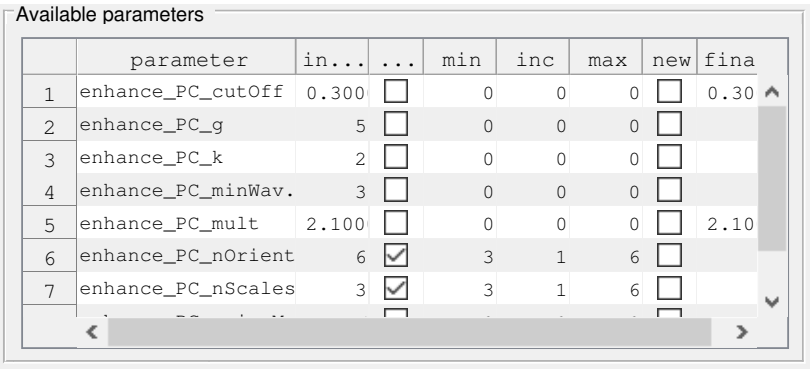


Figure 16.3: Check-boxes to select the enhancement algorithm



	parameter	in...	...	min	inc	max	new	fina
1	enhance_PC_cutOff	0.300	☐	0	0	0	☐	0.30
2	enhance_PC_g	5	☐	0	0	0	☐	
3	enhance_PC_k	2	☐	0	0	0	☐	
4	enhance_PC_minWav.	3	☐	0	0	0	☐	
5	enhance_PC_mult	2.100	☐	0	0	0	☐	2.10
6	enhance_PC_nOrient	6	☑	3	1	6	☐	
7	enhance_PC_nScales	3	☑	3	1	6	☐	

Figure 16.4: Table showing the parameters that can be tuned for the selected algorithm

the current value for each parameter is shown in the **initial** column, and cannot be edited. A minimum of two parameters must be selected using the **tune** check-boxes, and the range to explore set using the minimum value (**min**), the increment (**inc**), and the maximum value **max**. These two parameters are automatically set as the **X variable** and **Y variable** in the *Display* panel drop-down menus (Fig. 16.5). Each additional parameter selected for tuning, is displayed below the drop-down menus as a slider bar running between **Min** and **Max**. these slider are used to scroll through the images for display. In principle every parameter can be tuned in this way, but the number of factorial combinations will increase very rapidly, so it is wise to explore a more limited sub-set of parameters and values in the first instance, to progressively hone in on the best combination.

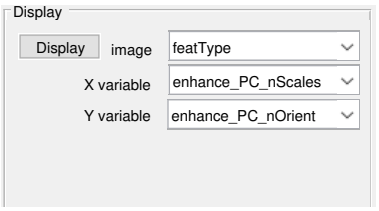


Figure 16.5: Controls to display a sub-set of the processed images

If a value for a parameter has already been optimised, the **new** check-box can be ticked in the *Available parameters* table, and the value entered in the **final** column.

When the **Process** button is clicked, the ROI will be processed using the factorial combination of the parameter range set.

Once the ROI has been processed, the results are displayed as a 2-D grid (Fig. 16.6), labelled with the values for the first two parameters selected for tuning in the *Display* panel. Different parameters can be selected for the display grid using the drop-down menus and sliders. The type of image displayed can be selected using the **image** drop-down menu in the *Display* panel.

At the end of the processing step, the images are also automatically segmented using the values set in the *Initial skeleton* panel.

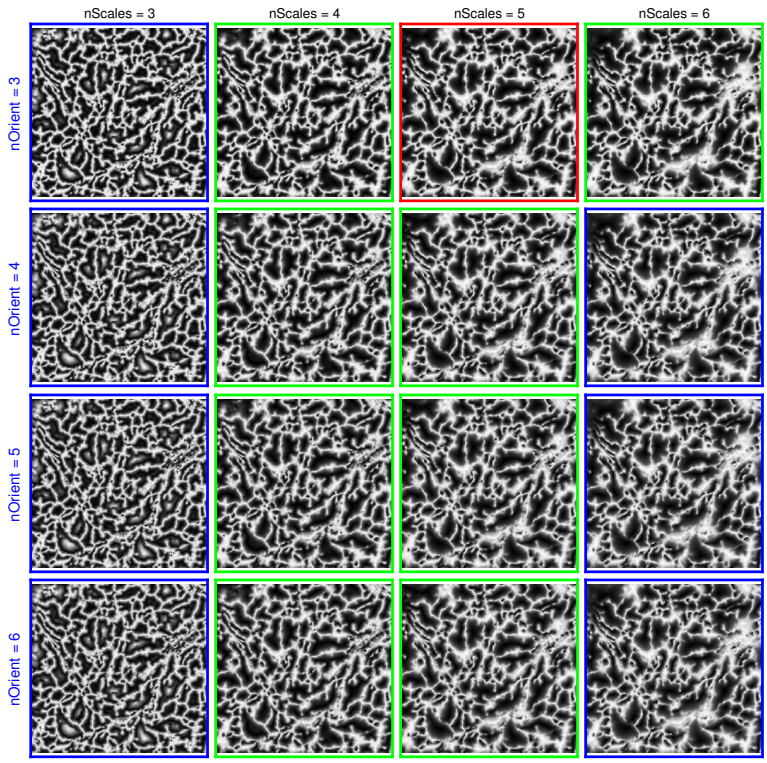


Figure 16.6: Grid showing results for the factorial sensitivity analysis for first two parameters

These follow the same options available in the main interface and process the enhanced image to a single-pixel wide skeleton. The skeleton can be displayed using the **image** drop-down menu. The skeletonisation parameters can be changed and just the skeletonisation step re-run on the set of enhanced images using the **Skeleton** button.

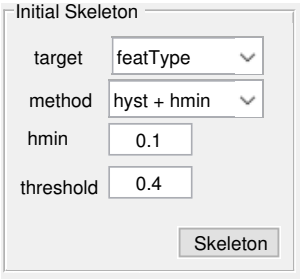


Figure 16.7: Initial segmentation controls

16.2 Comparison with a ground-truth skeleton

Pixel skeletons for each ridge enhancement-skeletonisation combination can be scored against a manually digitised ground-truth, within a chosen tolerance, typically around half the minimum vein width. Each pixel is classified as a true positive (TP), true negative (TN), false positive (FP), or false negative (FN). It is important that the ground-truth image is loaded into the main interface using the **Load GT** button (Fig. 16.8) prior to calling the *Parameter selector* GUI, so that the same ROI is extracted for comparison.

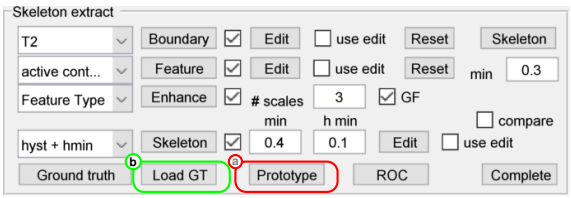


*Figure 16.8: The **Load GT** button in the skeleton extract panel*

The target enhanced image for comparison is selected using the **Target** drop-down menu in the *ROC analysis* panel, along with the segmentation **method**, and the **tolerance** (in pixels) that will still be considered as a 'true positive' relative to the Ground-Truth skeleton.

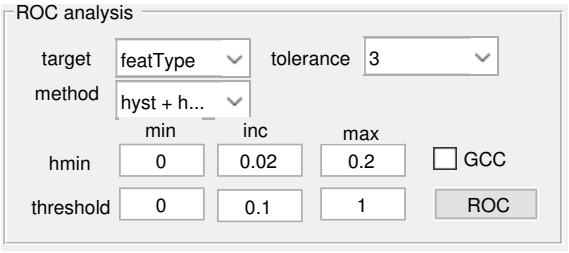


Figure 16.9: ROC analysis controls

If *hysteresis* thresholding is selected, the **threshold** options are enabled, and can also be set to run from **min** to **max**, with a given increment (**inc**). The analysis can be restricted to the giant connected component (GCC), using the **GCC** check-box, as a crude test of connectivity in the resultant skeleton. If the *WS + hmin* or *hyst + hmin* options is chosen, the **hmin** parameter is applied over a range of values given by the **min**, increment (**inc**), and **max** text boxes.

The **ROC** button initiates the analysis, and results are presented as a *ROC curve, P-R plot, z-score* or *histogram*, depending on the setting of the **ROC plot** drop-down menu in the *ROC plots* panel.

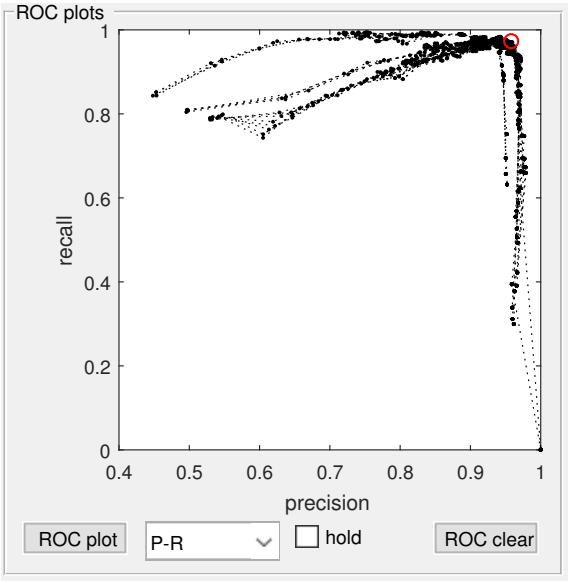


Figure 16.10: graphical output for a PRC curve

Precision-Recall (PRC) analysis is used in preference to Receiver Operating Characteristic (ROC) plots, as the former are better suited to imbalanced datasets [1], when the number of true negatives (TNs) from the background is expected to be much greater than the true positives (TPs) from the skeleton. Precision was calculated as TP/(TP+FP), and Recall as TP/(TP+FN).

Results for each run of the *hmin* or *threshold* parameter are connected by a dotted line. It should be noted that *hmin* and *threshold*

[1] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS One*, 10:e0118432, 2015

do not behave quite as a conventional tuneable threshold in ROC or PRC analysis, as the output (a connected skeleton) does not necessarily correlate linearily with increasing or decreasing values of the parameter. For example, as the threshold is lowered, one might expect fewer FNs, more TPs, but also more FPs. However, if the threshold merges two adjacent ridges, the resultant skeleton my be incorrectly positioned mid-way between both when the binary image is skeletonised. This give a decrease in FPs and and increase in FNs and FPs, and the overall performance may be worse than more stringent parameter settings

The **ROC clear** button erases the plot, whilst the **hold** check-box allows multiple analyses to be superimposed.

In addition to the PRC or ROC plot, the best performance can be estimated by a summary statistic selected from the *ROC summary* panel (Fig. 16.11).

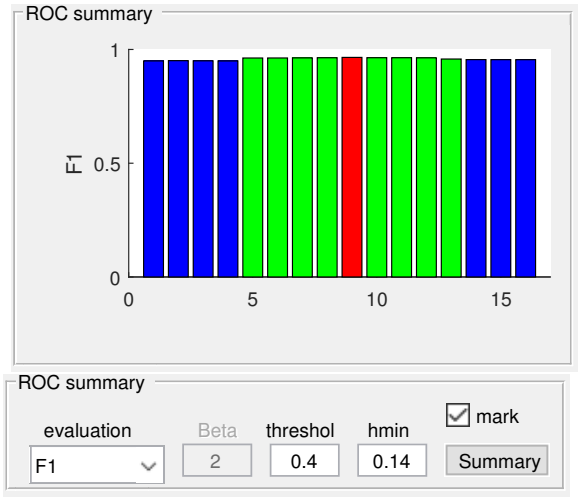


Figure 16.11: ROC summary controls and color-coded plot for the F_1 parameter

The best performance is assessed from the highest **evaluation** score, using a metric selected from the drop-down menu from *dprime*, F_1, *MCC*, F_2 or F_β. The F_β score represents a harmonic mean of recall and precision, where F_β, can be tuned to weight recall ($\beta = 2$) or precision ($\beta = 0.5$) more highly according to equation 16.1:

$$F_\beta = (1+\beta^2)\frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}} \tag{16.1}$$

The maximum score is colour-coded red on the histogram output, values within 98% coloured green and 95% coloured blue. If the **mark** check-box is ticked, the same colour coding is applied to the border of the corresponding image in the image grid. In addition, the best score is highlighted on the PRC or ROC plot with a red circle.

If multiple parameters have been selected for the sensitivity analysis, the image with the best response may not be visible in the current grid. In this case, the image can be revealed by scrolling through the different parameter values using the sliders in the

Display window. If the user is happy with the 'best' parameter combination, clicking on the preferred image will update the **final** parameter combination in the *Available parameters* table.

The *ROC* option in the *Display* panel, allows visualisation of the output of the ROC analysis in terms of TPs (green), FNs (red) and FPs (blue), for each of the parameter combinations visible (Fig. 16.12).

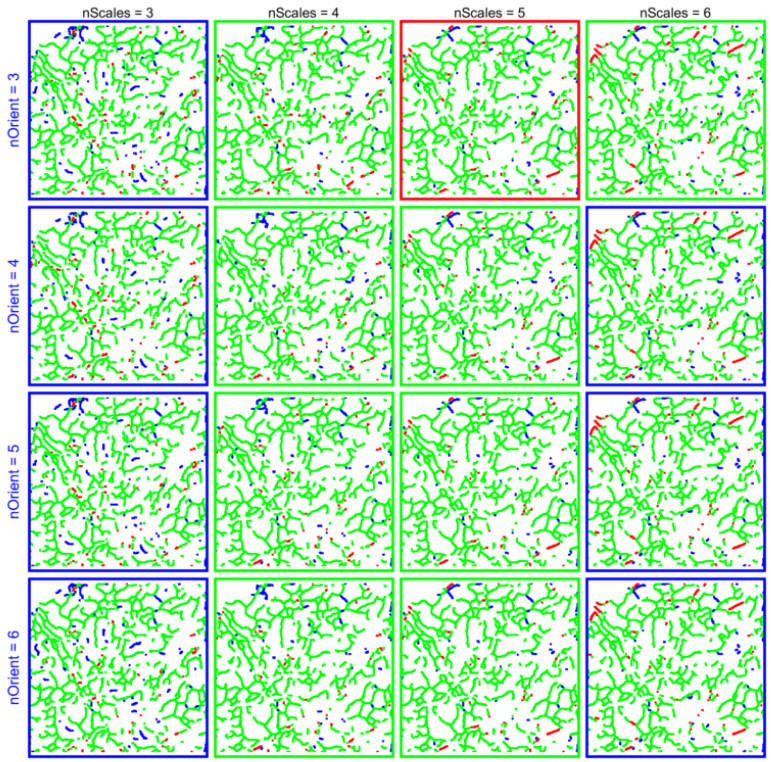


Figure 16.12: Grid showing TP (green), FN (red) and FP (blue) for skeletons derived from the first two parameters in the sensitivity analysis

16.3 Output

The current image grid can be saved using the **Images** button in the *Output* panel, whilst the complete set of summary statistics for each parameter combination can be saved to an Excel spreadsheet using the **Data** button. The **Panels** button saves each panel as a *.png and *.pdf image for inclusion in the manual.

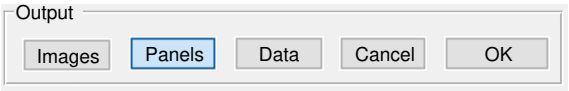


Figure 16.13: Output controls

When the *Parameter selector* GUI is closed the optimised parameters are returned to the main program. At this stage they are not automatically used to update the parameter settings to avoid accidentally corrupting the current parameters, thus if changes have

been made, they need to be explicitly changed using the **edit** button in the *Param* panel.

Bibliography

S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92:1–31, 2011.

J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994.

J. Bernsen. Dynamic thresholding of grey-level images. In *International conference on pattern recognition*, volume 2, pages 1251–1255, 1986.

A.-N. Bouchekhima, L. Frigerio, and M Kirkilionis. Geometric quantification of the plant endoplasmic reticulum. *J. Microscopy*, 234:158–172, 2009.

E. Breeze, N. Dzimitrowicz, V. Kriechbaumer, R. Brooks, S.W. Botchway, J.P. Brady, C. Hawes, A.M. Dixon, J.R. Schnell, M.D. Fricker, and L. Frigerio. A c-terminal amphipathic helix is necessary for the in vivo tubule-shaping function of a plant reticulon. *PNAS*, 113:10902–10907, 2016.

J. Buhl, J. Gautrais, R.V Solé, P. Kuntz, J.-L. Valverde, S.and Deneubourg, and G. Theraulaz. Efficiency and robustness in ant networks of galleries. *Eu. Phys. J. B*, 42:123–129, 2004.

T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10:266–277, 2001.

G. Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370, 2003.

A. F. Frangi, W. J. Niessen, K.L. Vincken, and M.A. Viergever. *Multiscale vessel enhancement filtering*, pages 130–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9: 891–906, 1991.

M.D. Fricker, D. Akita, L.L.M. Heaton, N. Jones, B. Obara, and T. Nakagaki. Automated analysis of physarum network structure and dynamics. *J. Phys. D*, 50:254005, 2017.

L. R. Griffing, C. Lin, C. Perico, R. R. White, and I. Sparkes. Plant ER geometry and dynamics: biophysical and cytoskeletal control during growth and biotic response. *Protoplasma*, 254:43–56, 2017.

P Haggett and RJ Chorley. *Network Analysis in Geography (pp 74-76) Edward Arnold Publishers Ltd*. London, 1969.

R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, pages 610–621, 1973.

C. Hawes, C.and Saint-Jore, B. Martin, and H.-Q. Zheng. Er confirmed as the location of mystery organelles in arabidopsis plants expressing gfp! *Trends in Plant Science*, 6:245–246, 2001.

K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1397–1409, 2013.

B. Hein, K. I. Willig, and S. W. Hell. Stimulated emission depletion (sted) nanoscopy of a fluorescent protein-labeled organelle inside a living cell. *PNAS*, 105:14271–14276, 2008.

B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17:185–203, 1981.

K. Kimura, A. Mamane, T. Sasaki, K. Sato, J. Takagi, R. Niwayama, Y. Hufnagel, L.and Shimamoto, J.-F. Joanny, S. Uchida, and A. Kimura. Endoplasmic-reticulum-mediated microtubule alignment governs cytoplasmic streaming. *Nat Cell Biol*, 19:399–406, 2017.

P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1:1–26, 1999.

P. Kovesi. Phase congruency: A low-level image invariant. *Psychological Research*, 64:136–148., 2000a.

P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing, 2000b. Available from: `http://www.peterkovesi.com/matlabfns/`.

D. J. Kroon, C. H. Slump, and T. J. Maal. Optimized anisotropic rotational invariant diffusion scheme on cone-beam ct. *Med Image Comput Comput Assist Interv*, 13:221–8, 2010.

T.J. Lambert and J.C. Waters. Navigating challenges in the application of superresolution microscopy. *J. Cell Biol.*, jcb.201610011, 2016.

V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Phys. Rev. Lett.*, 87:198701, 2001.

C. Lin, R.R. White, I. Sparkes, and P. Ashwin. Modeling endoplasmic reticulum network maintenance in a plant cell. *Biophysical Journal*, 113:214 – 222, 2017.

C. P. Lin, Y. W. Zhang, I. Sparkes, and P. Ashwin. Structure and dynamics of er: minimal networks and biophysical constraints. *Biophysical Journal*, 107:763–772, 2014.

M. Linkert, C. T. Rueden, C. Allan, J. M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. Macdonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. W. Eliceiri, and J. R. Swedlow. Metadata matters: access to image data in the real world. *J Cell Biol*, 189:777–82, 2010.

C. Lopez-Molina, G. V. D. de Ulzurrun, J. M. Baetens, J. Van den Bulcke, and B. De Baets. Unsupervised ridge detection using second order anisotropic gaussian kernels. *Signal Processing*, 116: 55–67, 2015.

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 674–679, 1981.

E. Meijering, M. Jacob, J. Sarria, P. Steiner, H. Hirling, and M Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry*, 58:167 – 176, 2004.

M.C. Morrone and R.A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303 – 313, 1987.

W. Niblack. *An Introduction to Image Processing*. Prentice-Hall, 1986.

C.J. Nixon-Abell, J.and Obara, A.V. Weigel, D. Li, W.R. Legant, C.S. Xu, H.A. Pasolli, K. Harvey, H.F. Hess, E. Betzig, C. Blackstone, and J. Lippincott-Schwartz. Increased spatiotemporal resolution reveals highly dynamic dense tubular matrices in the peripheral er. *Science*, 354:aaf3928, 2016.

B. Obara, V. Grau, and M. D. Fricker. A bioimage informatics approach to automatically extract complex fungal networks. *Bioinformatics*, 28:2374–81, 2012.

N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, Cyber.*, 9:62–66, 1979.

R.W. Ridge, Y. Uozumi, J. Plazinski, U.A. Hurley, and R.E. Williamson. Developmental transitions and dynamics of the cortical er of arabidopsis cells seen with green fluorescent protein. *Plant and Cell Physiology*, 40:1253–1261, 1999.

T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS One*, 10:e0118432, 2015.

J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33:225–236, 2000.

P.-L. Shui and W.-C. Zhang. Noise-robust edge detector combining isotropic and anisotropic gaussian kernels. *Pattern Recognition*, 45: 806 – 820, 2012.

I. Sparkes, T. Ketelaar, N.C.A. De Ruijter, and C. Hawes. Grab a Golgi: laser trapping of Golgi bodies reveals in vivo interactions with the endoplasmic reticulum. *Traffic*, 10:567–571, 2009a.

I. Sparkes, J. Runions, C. Hawes, and L. Griffing. Movement and remodeling of the endoplasmic reticulum in nondividing cells of tobacco leaves. *The Plant Cell*, 21:3937–3949, 2009b.

G. Stefano, L. Renna, and F. Brandizzi. The endoplasmic reticulum exerts control over organelle streaming during cell expansion. *J Cell Sci*, 127:947–53, 2014.

G. J. Streekstra and J. van Pelt. Analysis of tubular structures in three-dimensional confocal images. *Network: Computation in Neural Systems*, 13:381–395, 2002.

H. Ueda, E. Yokota, N. Kutsuna, T. Shimada, K. Tamura, T. Shimmen, S. Hasezawa, V. V. Dolja, and I. Hara-Nishimura. Myosin-dependent endoplasmic reticulum motility and f-actin organization in plant cells. *Proc Natl Acad Sci U S A*, 107:6894–9, 2010.

K. Ueda, H.and Tamura and I. Hara-Nishimura. Functions of plant-specific myosin xi: from intracellular motility to plant postures. *Curr. Op. Plant Biol.*, 28:30–38, 2015.

S. Venkatesh and R. Owens. On the classification of image features. *Pattern Recognition Letters*, 11:339–349, 1990.

L. M. Westrate, J. E. Lee, W. A. Prinz, and G. K. Voeltz. Form follows function: The importance of endoplasmic reticulum shape. *Annual Review of Biochemistry*, 84:791–811., 2015.

B. Zhang, J. Zerubia, and J.-C. (2007). Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Applied Optics*, 46:1819–1829, 2007.

K. Zuiderveld. *Contrast limited adaptive histograph equalization*, pages 474–485. Graphic Gems IV. Academic Press Professional, San Diego:, 1994.